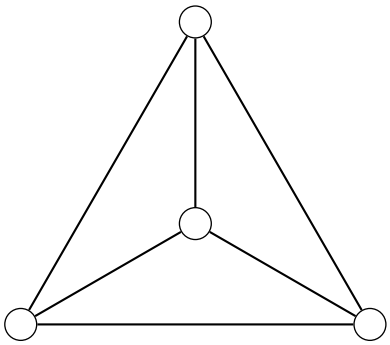
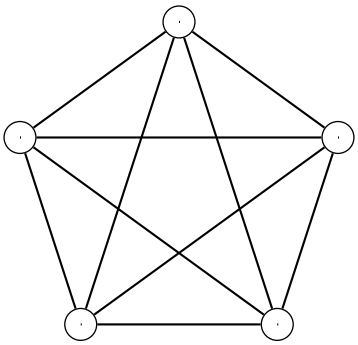
Gymnase de Burier 2023 – 2024

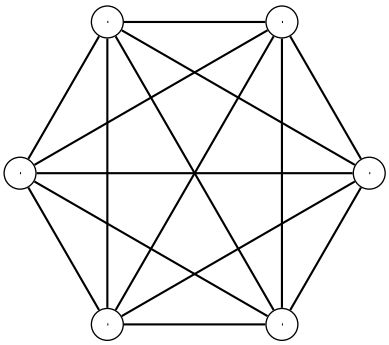
2OS – 3OS Mathématiques



K_4



K_5



K_6

Table des matières

Chapitre 1

Sommes et récurrence

1.1 Sommes

1.1.1 On donne

$$x_1 = 3, \quad x_2 = 5, \quad x_3 = 6, \quad x_4 = 2 \quad \text{et} \quad x_5 = 7.$$

Calculer :

a) $\displaystyle\sum_{i=1}^{5} x_i$

b) $\displaystyle\sum_{i=2}^{4} x_i$

c) $\displaystyle\sum_{k=1}^{5} x_k$

d) $\displaystyle\sum_{j=1}^{5} {x_j}^3$

e) $\displaystyle\sum_{i=1}^{5} (x_i + 8)$

f) $\displaystyle\sum_{k=1}^{5} (8 \cdot x_k)$

1.1.2 On donne

$$x_1 = 3, \quad x_2 = 5, \quad x_3 = 6, \quad x_4 = 2 \quad \text{et} \quad x_5 = 7.$$

On donne également

$$y_1 = 2, \quad y_2 = 8, \quad y_3 = 3, \quad y_4 = 1 \quad \text{et} \quad y_5 = 6.$$

Calculer :

a) $\displaystyle\sum_{i=1}^{5} (x_i + y_i)$

b) $\displaystyle\sum_{i=1}^{5} (x_i - y_i)$

c) $\displaystyle\sum_{k=1}^{5} (x_k \cdot y_k)$

d) $\displaystyle\sum_{j=1}^{5} (2 \cdot x_j)$

e) $\displaystyle\sum_{j=1}^{5} (x_j + y_j)^2$

f) $\displaystyle\sum_{i=1}^{5} 8$

g) $\displaystyle\sum_{i=1}^{5} (2 \cdot x_i) + \sum_{j=1}^{5} (3 \cdot y_j)$

h) $\displaystyle\sum_{j=1}^{5} {x_j}^2 - \left(\sum_{j=1}^{5} y_j \right)^2$

i) $\displaystyle\sum_{i=1}^{4} (x_{i+1} + y_i)$

1.1.3 Développer les sommes suivantes

a) $\sum_{i=0}^{n}(-1)^i$ b) $\sum_{j=0}^{n} 3j$ c) $\sum_{k=1}^{n} n$ d) $\sum_{i=1}^{n} \frac{n}{i}$

1.1.4 Développer chacune des sommes écrites à l'aide du symbole Σ, en faisant disparaître ce symbole :

a) $\sum_{k=3}^{10} \frac{1}{k^2}$ b) $\sum_{k=1}^{10} \frac{1}{2\,k+1}$ c) $\sum_{k=1}^{n} \frac{(k+1)!}{k}$

1.1.5 Développer et calculer les sommes suivantes :

a) $\sum_{i=0}^{n} 1$

b) $\sum_{j=1}^{2011}(-1)^j$

c) $\sum_{k=-n}^{n}(k+1)$

d) $\sum_{m=0}^{3}(m^2-6m+9)$

e) $\sum_{l=1}^{5} 4l\,(l^2-1)$

f) $\sum_{i=0}^{4}\left(2^i+\left(\frac{1}{2}\right)^i\right)$

g) $\sum_{k=1}^{5} \frac{k+2}{k}$

h) $\sum_{k=1}^{6} \frac{1}{3^k}$

i) $\sum_{j=-2}^{2} \frac{2^{j+3}}{j^2+1}$

1.1.6 Soit $x_1, \ldots, x_n$ une suite de nombres réels. Calculer $\sum_{i=2}^{n}(x_i - x_{i-1})$

1.1.7 Traduire à l'aide du symbole Σ les sommes suivantes :

a) $1^2+2^2+3^2+\cdots+12^2+13^2+14^2$

b) $11^2+12^2+13^2+\cdots+102^2+103^2+104^2$

c) $\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{13}{14}+\frac{14}{15}+\frac{15}{16}$

d) $1^2+3^2+5^2+\cdots+11^2+13^2+15^2$

e) $(1\cdot 3)+(2\cdot 4)+(3\cdot 5)+(4\cdot 6)+(5\cdot 7)$

f) $1+8+27+64+125$

1.1.8 Ecrire les sommes suivantes à l'aide du symbole Σ.

a) $2+4+6+\cdots 248 =$

b) $1000+1010+1020+\cdots+1540 =$

c) $1^2+2^2+3^2+\cdots 15^2 =$

d) $1+2+4+8+16+\cdots+1024 =$

e) $2+3+5+9+17+\cdots+1025 =$

f) $4+12+36+108+324 =$

g) $9-12+15-18+\cdots+303 =$

h) $45-40+35-30+25-20+15 =$

i) $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots+\frac{1}{n}=$

j) $\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{9}{10}+32=$

k) $\frac{1}{2}+\frac{4}{5}+\frac{9}{10}+\cdots+\frac{10000}{10001}=$

l) $\frac{5}{3}+\frac{10}{7}+\frac{15}{11}+\frac{20}{15}+\cdots+\frac{70}{55}=$

m) $(1\cdot3)+(2\cdot4)+(3\cdot5)+(4\cdot6)+(5\cdot7)=$

1.1.9 Traduire à l'aide du symbole Σ la somme suivante :

$$1000-\frac{3}{1\cdot(1+3+5)}+\frac{5}{(1+3)\cdot(1+3+5+7)}-\frac{7}{(1+3+5)\cdot(1+3+5+7+9)}+\ldots$$

1.1.10 Est-ce que $A=B=C=D$ si on définit

$$A=\sum_{i=1}^{n} i\cdot n \qquad B=\sum_{k=1}^{n} k\cdot n \qquad C=n\cdot\sum_{k=1}^{n} k \qquad D=k\cdot\sum_{i=0}^{n} n \qquad ?$$

1.1.11 Ecrire les sommes suivantes en faisant en sorte que la première valeur de l'indice soit 0, puis 49 :

a) $\sum_{i=2}^{45} 1=\sum_{i=0}^{\ldots}\ldots=\sum_{i=49}^{\ldots}\ldots$

b) $\sum_{i=10}^{20} i$

c) $\sum_{k=-4}^{180} \frac{k}{k+5}$

Faire calculer le résultat de chaque somme à l'aide d'une boucle `while` en python.

1.1.12 Changer l'indice et ses bornes de sorte que le terme général soit plus simple :

a) $\sum_{i=0}^{n} \frac{(i+1)^2+3}{1+\sqrt{i+1}}$ b) $\sum_{j=3}^{n+2} \frac{x^{j-3}}{(j-3)^x}$ c) $\sum_{k=1}^{n+1} (k-1)\,2^k+3^{k-1}$

1.1.13 On considère une suite de n nombres $x_1, x_2, x_3, \ldots, x_n$. Écrire les expressions suivantes de la manière la plus synthétique possible à l'aide d'un symbole de sommation.

a) La somme des termes de cette suite de nombres.

b) La somme des carrés des termes de cette suite de nombres

c) Le carré de la somme des termes de cette suite de nombres

d) La différence entre la somme des carrés et le carré de la somme des termes de cette suite de nombre

1.1.14 On considère deux suites de n nombres $x_1, x_2, x_3, \ldots, x_n$ et $y_1, y_2, y_3, \ldots, y_n$. Écrire les expressions suivantes de la manière la plus synthétique possible.

a) Produit de la somme des x_i et de la somme des y_i

b) Somme des produits $x_i\ y_i$

c) Somme des y_i moins k fois la somme des x_i

1.1.15 Ecrire un programme qui fait calculer les sommes suivantes :

a) $\sum_{i=1}^{100000} (2 + 3\ i)$

b) $\sum_{i=-237}^{325} i^2$

c) $\sum_{j=0}^{100} 2^j$

d) $\sum_{k=1}^{10^6} (2\,k - 1)$

1.1.16 Ecrire un programme qui fait calculer les sommes suivantes :

a) $\sum_{n=1}^{23} \sum_{k=1}^{37} k^n$

b) $\sum_{i=1}^{100} \sum_{j=1}^{100} 1$

c) $\sum_{n=1}^{20} \sum_{j=1}^{10} (n^j + 1)$

d) $\sum_{n=-10}^{10} \sum_{k=-n}^{n} k$

1.1.17 Soit $x_1, x_2, x_3, \ldots, x_n$ des nombres réels et $m = \frac{1}{n} \sum_{i=1}^{n} x_i$. Montrer que :

$$\frac{1}{n} \sum_{i=1}^{n} (x_i - m)^2 = \frac{1}{n} \sum_{i=1}^{n} {x_i}^2 - m^2$$

1.2 Récurrence

La démonstration par récurrence s'applique dans le cas où on doit démontrer une proposition P_n pour tout entier $n \geq 0$ ou $n \geq 1$ ou plus généralement $n \geq n_0$ ($n_0 \in \mathbb{N}$).

Une démonstration par récurrence se fait toujours en deux étapes :

a) Première phase, dite d'initialisation : on démontre que la proposition est vraie pour la plus petite valeur de l'entier n : 0, 1 ou n_0.

b) Deuxième phase : on suppose que la proposition P_n est vraie et on démontre sous cette hypothèse, appelée hypothèse de récurrence, que la proposition P_{n+1} est vraie.

D'un point de vue formel, on peut écrire :

$$\begin{cases} P_0 \text{ est vrai} \\ n \geq 0 \text{ et } P_n \Rightarrow P_{n+1} \end{cases} \Rightarrow \forall n \geq 0 \text{ : } P_n \text{ est vrai}$$

1.2.1 Démontrer par récurrence que, $\forall n \in \mathbb{N}^*$:

a) $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

b) $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

c) $\displaystyle\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$

d) $\displaystyle\sum_{k=1}^{n} (-1)^{k+1}\ k^2 = (-1)^{n+1}\ \frac{n(n+1)}{2}$

e) $\displaystyle\sum_{k=1}^{n} \frac{1}{(2\,k-1)\,(2\,k+1)} = \frac{n}{2\,n+1}$

f) $\displaystyle\sum_{k=1}^{n} \frac{k^2}{(2\,k-1)\,(2\,k+1)} = \frac{n\ (n+1)}{2\ (2\,n+1)}$

g) $\displaystyle\sum_{k=1}^{n} \frac{k}{2^k} = 2 - \frac{n+2}{2^n}$

h) $\displaystyle\sum_{k=1}^{n} k \cdot 5^k = \frac{5 + (4\,n-1)\ 5^{n+1}}{16}$

i) $\displaystyle\sum_{k=1}^{n} \frac{1}{k\ (k+1)\,(k+2)} = \frac{n\ (n+3)}{4\ (n+1)\,(n+2)}$

1.2.2 Démontrer par récurrence que, $\forall n \in \mathbb{N}$:

a) $8^n - 1$ est divisible par 7

b) $3^{2\,n+2} - 2^{n+1}$ est un multiple de 7

c) $10^{6\,n+2} + 10^{3\,n+1} + 1$ est un multiple de 111

d) $n^3 + 5\,n$ est divisible par 3

e) $\frac{2}{3}\,n^3 + n^2 + \frac{1}{3}\,n$ est un nombre pair

f) $11^{n+2} + 12^{2\,n+1}$ est divisible par 133

1.2.3 Soit $n \in \mathbb{N}$. Prouver par récurrence puis directement que 3 divise $n^3 - n$.

1.2.4 Établir une formule pour la somme

$$\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \cdots + \frac{1}{n\ (n+1)}$$

1.2.5 Établir une formule pour la somme

$$1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \cdots + \frac{1}{1+2+3+\cdots+n}$$

1.2.6 Écrire en fonction de n la somme des carrés des n premiers nombres impairs.

1.2.7 Démontrer que « $\forall n \in \mathbb{N},\ n^2 - n + 41$ *est premier* » est une proposition fausse.

1.2.8 Démontrer que « $\forall n \in \mathbb{N}^*,\ (n$ *est premier* $\Rightarrow 2^n - 1$ *est premier* $)$ » est une proposition fausse.

1.2.9 Trouver l'erreur commise dans la démonstration ci-dessous :
On note P_n l'assertion « $\forall n \in \mathbb{N}^*,\ (6$ *divise* $7^n + 1)$ ».

Prouvons que 6 divise 7^n+1. Supposons que P_n est vraie. Alors il existe k tel que $7^n+1 = 6\,k$. Ainsi, $7^{n+1}+1 = 7\cdot 7^n+1 = 7\,(6\,k-1)+1 = 42\,k-6 = 6\,(7\,k-1)$. Par conséquent 6 divise $7^{n+1}+1$, i.e. P_{n+1} est vraie. Par récurrence, P_n est vraie pour tout n.

1.2.10 Trouver l'erreur commise dans la démonstration ci-dessous :
Prouvons que tout le monde a la même taille. On pose $P_n =$ « n personnes ont toujours la même taille ». Cette fois on n'oublie pas d'initialiser : une personne a la même taille qu'elle-même, donc P_1 est vraie.
L'hérédité ensuite. Soit $n > 1$ et supposons que P_n est vraie. On considère alors $n+1$ personnes. Les personnes 1 jusqu'à n ont la même taille, par hypothèse de récurrence. De même, les personnes 2 jusqu'à $n+1$ ont la même taille. Ainsi, ces $n+1$ personnes ont la même taille.

1.2.11 Démontrer par récurrence :

a) Pour tout nombre réel positif x, la propriété suivante est vérifiée :
$\forall n \in \mathbb{N}, (1+x)^n \geq 1+n\,x$

b) Pour tout $a \in \mathbb{R}$ et $r \in \mathbb{R} - \{1\}$, on a :
$$\forall n \in \mathbb{N}^*, \quad a + a\,r + a\,r^2 + \cdots + a\,r^{n-1} = \frac{a\,(r^n-1)}{r-1}$$

c) Pour tout $\alpha \in \mathbb{R} - \{k\,\pi \,|\, k \in \mathbb{Z}\}$, on a :
$$\forall n \in \mathbb{N}, \quad \cos(\alpha)\cdot\cos(2\,\alpha)\cdot\cos(4\,\alpha)\cdot\ldots\cdot\cos(2^n\,\alpha) = \frac{\sin(2^{n+1}\,\alpha)}{2^{n+1}\,\sin(\alpha)}$$

d) Pour des nombres réels strictement positifs $x_1, x_2, x_3, \ldots$, on a :
$$\forall n \in \mathbb{N}, \quad (x_1+x_2+\cdots+x_n)\cdot\left(\frac{1}{x_1}+\frac{1}{x_2}+\ldots+\frac{1}{x_n}\right) \geq n^2$$

1.2.12 Démontrer par récurrence :

a) $\forall n \in \mathbb{N}^*, \quad (n+1)! > 1! + 2! + 3! + \cdots + n!$

b) $\forall n \in \mathbb{N} - \{0\,,\,1\,,\,2\,,\,3\}, \quad n! > 2^n$

1.2.13 Soit (u_n) la suite de *Fibonacci* définie comme suit :

$$\begin{cases} u_1 = 1 \\ u_2 = 1 \\ u_{n+1} = u_n + u_{n-1}, \quad \text{pour} \quad n \geq 2 \end{cases}$$

a) Démontrer que $\forall n \in \mathbb{N}^* \quad , \quad {u_1}^2 + {u_2}^2 + \ldots + {u_n}^2 = u_n \cdot u_{n+1}$

b) Calculer l'expression ${u_n}^2 - u_{n-1}\cdot u_{n+1}$ pour quelques valeurs de n.

c) Démontrer que $\forall n \in \mathbb{N}^* \quad , \quad u_n = \dfrac{\left(1+\sqrt{5}\right)^n - \left(1-\sqrt{5}\right)^n}{\sqrt{5}\cdot 2^n}$

1.3 Solutions des exercices

1.1.1

a) $\sum_{i=1}^{5} x_i = 23$

b) $\sum_{i=2}^{4} x_i = 13$

c) $\sum_{k=1}^{5} x_k = 23$

d) $\sum_{j=1}^{5} {x_j}^3 = 719$

e) $\sum_{i=1}^{5} (x_i + 8) = 63$

f) $\sum_{k=1}^{5} (8 \cdot x_k) = 184$

1.1.2

a) $\sum_{i=1}^{5} (x_i + y_i) = 43$

b) $\sum_{i=1}^{5} (x_i - y_i) = 3$

c) $\sum_{k=1}^{5} (x_k \cdot y_k) = 108$

d) $\sum_{j=1}^{5} (2 \cdot x_j) = 46$

e) $\sum_{j=1}^{5} (x_i + y_i)^2 = 453$

f) $\sum_{i=1}^{5} 8 = 40$

g) $\sum_{i=1}^{5} (2 \cdot x_i) + \sum_{j=1}^{5} (3 \cdot y_j) = 106$

h) $\sum_{j=1}^{5} {x_j}^2 - \left(\sum_{j=1}^{5} y_j \right)^2 = -277$

i) $\sum_{i=1}^{4} (x_{i+1} + y_i) = 34$

1.1.3

a) 1 si n est pair ; 0 si n est impair

b) $3 + 6 + 9 + \ldots + 3n$

c) n^2

d) $n + \frac{n}{2} + \frac{n}{3} + \ldots + \frac{n}{n-1} + 1$

1.1.4 –

1.1.5

a) $n + 1$

b) -1

c) $2\,n+1$

d) $\sum_{m=0}^{3}(m-3)^2 = 14$

e) $4\cdot 0 + 8\cdot 3 + 12\cdot 8 + 16\cdot 15 + 20\cdot 24 = 840$

f) $1+1+2+\frac{1}{2}+4+\frac{1}{4}+8+\frac{1}{8}+16+\frac{1}{16} = \frac{527}{16}$

g) $3+2+\frac{5}{3}+\frac{3}{2}+\frac{7}{5} = \frac{287}{30}$

h) $\frac{1}{3}+\frac{1}{9}+\frac{1}{27}+\frac{1}{81}+\frac{1}{243}+\frac{1}{729} = \frac{364}{729}$

i) $\frac{2}{5}+\frac{4}{2}+\frac{8}{1}+\frac{16}{2}+\frac{32}{5} = \frac{124}{5}$

1.1.6 $\sum_{i=2}^{n}(x_i - x_{i-1}) = x_n - x_1$

1.1.7

a) $1^2+2^2+3^2+\cdots+12^2+13^2+14^2 = \sum_{i=1}^{14} i^2 = \sum_{i=0}^{13}(i+1)^2$

b) $11^2+12^2+13^2+\cdots+102^2+103^2+104^2 = \sum_{i=11}^{104} i^2$

c) $\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{13}{14}+\frac{14}{15}+\frac{15}{16} = \sum_{i=1}^{15}\frac{i}{i+1}$

d) $1^2+3^2+5^2+\cdots+11^2+13^2+15^2 = \sum_{i=1}^{8}(2\,i-1)^2$

e) $(1\cdot 3)+(2\cdot 4)+(3\cdot 5)+(4\cdot 6)+(5\cdot 7) = \sum_{i=1}^{5} i\,(i+2)$

f) $1+8+27+64+125 = \sum_{i=1}^{5} i^3$

1.1.8

a) $\sum_{i=1}^{124} 2i$

b) $\sum_{i=100}^{154} 10i$

c) $\sum_{k=1}^{15} k^2$

d) $\sum_{i=0}^{10} 2^i$

e) $\sum_{i=0}^{10}(2^i+1)$

f) $\sum_{i=0}^{4} 4\cdot 3^i$

g) $\sum_{i=3}^{101}(-1)^{i+1}\,3i$

h) $\sum_{i=0}^{6}(-1)^i(45-5i)$

i) $\sum_{i=1}^{n}\frac{1}{i}$

j) $\sum_{i=1}^{9}\frac{i}{i+1}+32$

k) $\sum_{i=1}^{100}\frac{i^2}{i^2+1}$

l) $\sum_{i=1}^{14}\frac{5i}{4i-1}$

m) $\sum_{j=1}^{5} j(j+2)$

1.1.9 $1000+\sum_{i=1}^{\infty}\frac{(-1)^i(2i+1)}{\sum_{j=1}^{i}(2j-1)\sum_{k=1}^{i+2}(2k-1)}$

1.1.10 $A=B=C=\frac{n^2(n+1)}{2}\ \neq D=k\,n\,(n+1)$

1.1.11

a) $\sum_{i=2}^{45}1=\sum_{i=0}^{43}1=\sum_{i=49}^{92}1$

b) $\sum_{i=10}^{20} i=\sum_{i=0}^{10}(i+10)=\sum_{i=49}^{59}(i-39)$

c) $\sum_{k=-4}^{180}\frac{k}{k+5}=\sum_{k=0}^{184}\frac{k-4}{k+1}=\sum_{k=49}^{233}\frac{k-53}{k-48}$

1.1.12

a) $\sum_{i=0}^{n}\frac{(i+1)^2+3}{1+\sqrt{i+1}}=\sum_{j=1}^{n+1}\frac{j^2+3}{1+\sqrt{j}}$

b) $\sum_{j=3}^{n+2}\frac{x^{j-3}}{(j-3)^x}=\sum_{k=0}^{n-1}\frac{x^k}{k^x}$

c) $\sum_{k=1}^{n+1}(k-1)\,2^k+3^{k-1}=\sum_{l=0}^{n} l\cdot 2^{l+1}+3^l$

1.1.13

a) $\sum x_i$

b) $\sum(x_i)^2=\sum {x_i}^2$

c) $\left(\sum x_i\right)^2$

d) $\sum {x_i}^2-\left(\sum x_i\right)^2$

1.1.14

a) $\left(\sum_i x_i\right)\left(\sum_i y_i\right)=\sum_i x_i\ \sum_j y_j=\sum_{i,j} x_i y_j$

b) $\sum_i x_i\,y_i$

c) $\sum y_i-k\sum x_i$

1.1.15

```
def somme(f, n, m):
    s = 0
    for i in range(n, m + 1):
        s += f(i)
    return s
def a(i):
    return 2 + 3 * i
def b(i):
    return i**2
def c(j):
    return 2**j
def d(k):
    return 2*k - 1


print("a)", somme(a, 1, 100000))
print("b)", somme(b, -237, 325))
print("c)", somme(c, 0, 100))
print("d)", somme(d, 1, 10**6))
```

1.1.16

```
def somme(f, n_1, m_1, n_2, m_2):
    s = 0
    for i in range(n_1, m_1 + 1):
        for j in range(n_2, m_2 + 1):
            s += f(i, j)
    return s
def a(k, n):
    return k^n
def b(i, j):
    return 1
```

```
def c(n, j):
    return n**j + 1


print("a)", somme(a, 1, 23, 1, 37))
print("b)", somme(b, 1, 100, 1, 100))
print("c)", somme(c, 1, 20, 1, 10))


d = 0


for n in range(-10, 11):
    for k in range(-n, n + 1):
        d += k


print("d)", d)
```

1.1.17

1.2.1 –

1.2.2 –

1.2.3 –

1.2.4 –

1.2.5 –

1.2.6 –

1.2.7 –

1.2.8 –

1.2.9 –

1.2.10 –

1.2.11 –

1.2.12 –

1.2.13 –

Chapitre 2

Cryptologie

2.1 Chiffrement polyalphabétique

2.1.1 (Chiffre de Vigenère)

a) Chiffrer à l'aide du chiffrement de Vigenère le texte suivant : `TEXTESECRETADECODER` en utilisant comme clé le mot `CRYPTO`.

b) Pour le même texte clair on obtient le texte chiffré suivant `BRQKSMZCSPXIQXTCXZR`. Quelle est la clé ?

c) Déchiffrer le message `VHOSDAVH` avec la clé `CAKE`.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
BCDEFGHIJKLMNOPQRSTUVWXYZA
CDEFGHIJKLMNOPQRSTUVWXYZAB
DEFGHIJKLMNOPQRSTUVWXYZABC
EFGHIJKLMNOPQRSTUVWXYZABCD
FGHIJKLMNOPQRSTUVWXYZABCDE
GHIJKLMNOPQRSTUVWXYZABCDEF
HIJKLMNOPQRSTUVWXYZABCDEFG
IJKLMNOPQRSTUVWXYZABCDEFGH
JKLMNOPQRSTUVWXYZABCDEFGHI
KLMNOPQRSTUVWXYZABCDEFGHIJ
LMNOPQRSTUVWXYZABCDEFGHIJK
MNOPQRSTUVWXYZABCDEFGHIJKL
NOPQRSTUVWXYZABCDEFGHIJKLM
OPQRSTUVWXYZABCDEFGHIJKLMN
PQRSTUVWXYZABCDEFGHIJKLMNO
QRSTUVWXYZABCDEFGHIJKLMNOP
RSTUVWXYZABCDEFGHIJKLMNOPQ
STUVWXYZABCDEFGHIJKLMNOPQR
```

```
TUVWXYZABCDEFGHIJKLMNOPQRS
UVWXYZABCDEFGHIJKLMNOPQRST
VWXYZABCDEFGHIJKLMNOPQRSTU
WXYZABCDEFGHIJKLMNOPQRSTUV
XYZABCDEFGHIJKLMNOPQRSTUVW
YZABCDEFGHIJKLMNOPQRSTUVWX
ZABCDEFGHIJKLMNOPQRSTUVWXY
```

2.1.2 Qu'apporte le chiffrement de Vigenère en matière de sécurité par rapport à une simple substitution monoalphabétique ?

2.1.3

a) Chiffrer à l'aide de l'algorithme de Vigenère le texte suivant : `TEXTESECRETADECODER` en utilisant comme clé le mot `CRYPTO`.

b) Pour le même texte clair on obtient le texte chiffré suivant

`BRQKSMZCSPXIQXTCXZR`

Quelle est la clé ?

2.1.4 (Chiffre de Polybe) On considère l'alphabet privé du `W`, soit 25 lettres. `Polybe` a proposé le mécanisme de chiffrement suivant : on range les lettres dans un tableau `5 X 5`, en commençant par le mot clé (et en supprimant les doublons), puis on continue avec les lettres restantes de l'alphabet, dans l'ordre.
Par exemple, avec le mot-clé `MYSTERE`, on construit le tableau suivant :

	1	2	3	4	5
1	M	Y	S	T	E
2	R	A	B	C	D
3	F	G	H	I	J
4	K	L	N	O	P
5	Q	U	V	X	Z

Le chiffrement s'effectue alors en remplaçant chaque lettre par les deux chiffres : ligne colonne qui indiquent sa position dans la grille.
Par exemple, F est chiffré 31.

Ce chiffrement est-il polyalphabétique ?

Déchiffrer le message suivant

```
421513 132243 324244 141342 444332 132515 135334 444244
431325 154222 521444 114315 234215 131315 431411 444324
441552 212552 431542 224332 521552 211144 434414 444315
```

2.1.5 Écrire les fonctions python nécessaire pour chiffrer et déchiffrer un message secret en utilisant le chiffrement dit de Vigenère.

2.1.6 Écrire les fonctions python nécessaire pour chiffrer et déchiffrer un message secret en utilisant le chiffrement dit à clef progressive.

2.1.7 Écrire les fonctions python nécessaire pour chiffrer et déchiffrer un message secret en utilisant le chiffrement dit autoclave.

2.2 Plus grand diviseur commun et plus petit multiple commun

2.2.1 Soit a et b deux entiers positifs. On note $\gcd(a, b)$, de l'anglais *greatest common divisior*, le plus grand diviseur commun de a et b. On peut aussi le noter $\mathrm{pgcd}(a, b)$.

Montrer que $\gcd(a, b)$ existe.

On dira de plus que a et b sont *premiers entre eux* si $\gcd(a, b) = 1$

2.2.2 Trouver tous les diviseurs communs positifs de

a) 16 et 48,

b) 30 et 45,

c) 18 et 65.

2.2.3 Trouver le plus grand diviseur commun de

a) 35 et 65,

b) 135 et 156,

c) 49 et 99.

2.2.4 Trouver le plus grand diviseur commun de 17017 et 19210.

2.2.5 Trouver le plus grand diviseur commun de de 21331 et de 43947. (L'utilisation d'une calculette peut être judicieuse.)

2.2.6 Trouver le plus grand diviseur commun de de 210632 et de 423137. (L'utilisation d'un ordinateur peut être judicieuse.)

2.2.7 Montrer que pour tout $n \in \mathbb{N}$, n et $n + 1$ sont premiers entre eux.

2.2.8 Montrer que si $a \mid b$, alors $\gcd(a, b) = a$.

2.2.9 Soit a et b deux nombres entiers. Supposons qu'il existe r et s deux nombres, également entiers, tels que $a \cdot r + b \cdot s = 1$. Montrer que a et b sont premiers entre eux.

2.3 Algorithme d'Euclide

2.3.1 Appliquer l'algorithme d'Euclide (soustractions successives) à

a) 135 et 156 ;

b) 17017 et 19210 ;

c) 21331 et 43947.

2.3.2 Appliquer l'algorithme d'Euclide (divisions avec reste) à

a) 121 et 365 ;

b) 89 et 144 ;

c) 295 et 595 ;

d) 1001 et 1309.

2.3.3 En utilisant l'algorithme d'Euclide (divisions avec reste), trouver le plus grand diviseur commun de

a) 17017 et 18900 ;

b) 21063 et 43137 ;

c) 92263 et 159037 ;

d) 112345 et 112354.

2.3.4 Montrer que si e est un diviseur de a et que e est un diviseur de b, alors e est un diviseur de $a\,r + b\,s$ pour des entiers r et s quelconques.

2.3.5 Calculer $\mathrm{pgcd}(987, 610)$ à l'aide de l'énumération des diviseurs, puis à l'aide de l'algorithme d'Euclide.

2.3.6 Calculer à l'aide de l'algorithme d'Euclide le pgcd des nombres a et b suivants :

a) $a = 1233$, $b = 9999$;

b) $a = 12345$, $b = 54321$;

2.3.7 Calculer les nombres s et t tels que $s \cdot a + t \cdot b = \mathrm{pgcd}(a, b)$, avec les nombres a et b suivants :

a) $a = 72$, $b = 39$;

b) $a = 1008$, $b = 25$;

c) $a = 1993$, $b = 210$;

d) $a = 1995$, $b = 323$.

2.3.8 Montrer que $\mathrm{pgcd}(n, 0) = n$, quel que soit l'entier naturel n.

2.3.9 Écrire un programme qui calcule le pgcd de deux nombres entiers a et b à l'aide de l'algorithme d'Euclide.

2.4 Algorithme d'Euclide étendu

2.4.1 En utilisant l'algorithme d'Euclide étendu, trouver d, le plus grand diviseur commun et trouver également r, s tels que $a\,r + b\,s = d$ pour a et b donnés ci-dessous :

a) 270 et 114 ;

b) 242 et 1870 ;

c) 600 et 11312 ;

d) 11213 et 1001 ;

e) 500 et 3000.

2.4.2 Écrire un programme qui permet de calculer à l'aide de l'algorithme d'Euclide étendu les nombres s et t tels que $a \cdot s + b \cdot t = \mathrm{pgcd}(a, b)$, pour a et b deux entiers positifs quelconques.
Utiliser ce programme pour calculer s, t et $\mathrm{pgcd}(a, b)$ si a et b valent :

a) 3405 et 3367 ;

b) 13594140393 et 542988845 ;

c) 17192503173298665204451 et 3719762684395467374891 ;

d) 83575760507228709062535623 et 125120843342028658261747445 ;

e) 16526872358022374282401702663316163965480415756429 63 et 11348642810855305786834316686113495491269993843097 21.

2.4.3 Montrer que, pour tout nombre naturel n, les nombres n et $n^2 + 1$ sont premiers entre eux.

2.5 Exponentiation modulaire

2.5.1 Si n est un nombre naturel, la n-ième puissance d'un nombre a est, par définition, le produit de n facteurs égaux à a. Ainsi, d'après cette définition, le calcul de a^n nécessite $n-1$ multiplications. On peut cependant obtenir le même résultat en effectuant moins d'opérations.
Voici à titre d'exemple l'évaluation de a^{35}.

— On écrit l'exposant n comme une somme de puissance de 2. Ici, $35 = 32 + 2 + 1$;

— on calcule ensuite les puissances paires de a : $a^2 = a \cdot a$, $a^4 = a^2 \cdot a^2$, $a^8 = a^4 \cdot a^4$, $a^{16} = a^8 \cdot a^8$, $a^{32} = a^{16} \cdot a^{16}$.

— on multiplie pour terminer les « bons » carrés : $a^{35} = a^{32} \cdot a^2 \cdot a^1$;
Le nombre de multiplications nécessaires est dans ce cas de 7, au lieu de 34.

a) Combien de multiplications nécessite cet algorithme pour calculer chacune des puissances suivantes : a^{10}, a^{61}, a^{1000} ?

b) Calculer $835^{25} \pmod{1073}$, en 6 multiplications.

2.5.2 Dans le shell de python, calculer

$$17^{13} \mod 23$$

en suivant la procédure ci-dessous :

a) Stocker la valeur de l'exposant dans une variable : `n = 13`.

b) Déclarer deux variables `r` et `x`, dont la valeur initiale est 1 et 17, respectivement.

c) À l'aide de la fonction prédéfinie `bin`, obtenir le code binaire de l'exposant : `bin(n)`.

d) En utilisant judicieusement et le nombre de fois que c'est nécessaire les instructions ci-dessous, faire en sorte que la variable `r` contienne le résultat cherché.

```
r = r*x % 23
x = x*x % 23
```

2.5.3 En utilisant une seule boucle `while`, des divisions entières par 2 et des calculs de restes modulo 2, écrire en python le code qui fait afficher à la console le code binaire d'un nombre entier, sur une colonne.

2.5.4 Effectuer les calculs ci-dessous :

a) $25^5 \mod 133$

b) $100^{65} \mod 133$

c) $107^5 \mod 133$

d) $46^{65} \mod 133$

e) $234^{65} \mod 667$

f) $447^{65} \mod 667$

g) $99^{417} \mod 667$

h) $664^{65} \mod 667$

2.5.5 Écrire un programme en python qui permet de faire calculer

$$a^b \mod n$$

pour a, b et n des nombres entiers positifs qui sont formés de centaines de chiffres.

2.5.6 Utiliser le programme de l'exercice précédent pour faire calculer $a^b \mod n$, si

$a = 120394871029348710928374019238472034923495602345762938476565823$

$b = 202932093845709283475092834750928347509283475092384750923847507$

$n = 323490850293452304958775209384776108273649876450247856029384751$

2.6 Théorèmes de Fermat et d'Euler

Soit $m \geq 2$ et a un entier tel que $\text{pgcd}(m, a) = 1$. L'*ordre de a modulo m* est le plus petit entier positif e tel que $a^e \equiv 1 \mod m$.

2.6.1 Déterminer l'ordre de chaque élément non nul de $\mathbb{Z}_5$.

2.6.2 Déterminer l'ordre de chaque inversible de $\mathbb{Z}_9$.

2.6.3 Trouver l'ordre de 2 dans $\mathbb{Z}_m$ si

a) $m = 11$;

b) $m = 17$;

c) $m = 31$;

d) $m = 9$;

e) $m = 14$.

2.6.4 Ecrire un programme en python qui permet de trouver l'ordre de a dans $\mathbb{Z}_m$, sachant que $\text{pgcd}(m, a) = 1$.

2.6.5 À l'aide du programme écrit à l'exercice précédent, trouver l'ordre de tous les éléments non nuls de $\mathbb{Z}_m$ si

a) $m = 11$;

b) $m = 13$;

c) $m = 17$.

2.6.6 À l'aide du même programme, trouver l'ordre de tous les éléments inversibles de $\mathbb{Z}_{24}$

2.6.7 Effectuer les calculs ci-dessous et réduire le résultat modulo 7.

a) 5^6 b) 2^6 c) 4^6 d) 6^6

2.6.8 Effectuer les calculs ci-dessous et réduire le résultat modulo 17.

a) 8^{16} b) 2^{16} c) 11^{16} d) 5^{16}

2.6.9 Effectuer les calculs ci-dessous :

a) $3^5 \mod 5$ d) $5^7 \mod 7$ g) $11^{13} \mod 13$ j) $8^{19} \mod 19$

b) $2^5 \mod 5$ e) $6^7 \mod 7$ h) $10^{19} \mod 19$ k) $12^{17} \mod 17$

c) $3^7 \mod 7$ f) $5^{13} \mod 13$ i) $2^{19} \mod 19$ l) $10^{17} \mod 17$

Que peut-on en déduire ?

2.6.10 À l'aide de calculs élémentaires et sans utiliser directement le théorème de Fermat, expliquer pourquoi si a est inversible dans $\mathbb{Z}_{15}$, alors

$$a^8 \equiv 1 \mod 15$$

2.6.11 De même, expliquer pourquoi si a est inversible dans $\mathbb{Z}_{21}$, alors

$$a^{12} \equiv 1 \mod 21$$

2.6.12 À l'aide du théorème de Fermat, calculer $2^9 \mod 11$ et vérifier que 2^9 est l'inverse de 2 modulo 11.

2.6.13 On note $\varphi(n)$ le nombre d'éléments inversibles de $\mathbb{Z}_n$. Calculer $\varphi(n)$ pour les valeurs de n données ci-dessous :

a) $n = 4$, $n = 8$, $n = 16$, $n = 32$;

b) $n = 3$, $n =9$, $n = 27$, $n = 81$;

c) $n = 7^5$;

d) $n = 4027$;

e) $n = 4087$;

2.6.14 Écrire, en python, un programme qui fait calculer $\varphi(m)$, pour m un entier positif donné.

2.6.15 Vérifier le théorème d'Euler pour 1, 3, 7 et 9 dans $\mathbb{Z}_{10}$.

2.6.16 Calculer $a^5 \mod 7$ pour $a = 1, 2, 3, 4, 5, 6$ et vérifier que

$$a^5 \cdot a \equiv 1 \mod 7$$

2.6.17 Vérifier l'équivalence ci-dessous

$$2^{\varphi(21)} \equiv 1 \mod 21$$

2.6.18 Calculer $48^{322} \mod 25$.

2.6.19 Calculer $40^{322} \mod 21$.

2.7 Factoriser un nombre entier

2.7.1 Un nombre supérieur à 1 est dit *premier* s'il a exactement deux diviseurs : 1 et lui-même. Les nombres qui ne sont pas premiers sont dits *composés*.

Soit $n \in \mathbb{N}$ un nombre composé. Montrer que n admet un facteur inférieur ou égal à $\sqrt{n}$.

2.7.2 A l'aide du crible d'Eratosthène, déterminer « à la main » les nombres premiers inférieurs à 500.

2	3	5	7	9	11	13	15	17	19
21	23	25	27	29	31	33	35	37	39
41	43	45	47	49	51	53	55	57	59
61	63	65	67	69	71	73	75	77	79
81	83	85	87	89	91	93	95	97	99
101	103	105	107	109	111	113	115	117	119
121	123	125	127	129	131	133	135	137	139
141	143	145	147	149	151	153	155	157	159
161	163	165	167	169	171	173	175	177	179
181	183	185	187	189	191	193	195	197	199
201	203	205	207	209	211	213	215	217	219
221	223	225	227	229	231	233	235	237	239
241	243	245	247	249	251	253	255	257	259
261	263	265	267	269	271	273	275	277	279
281	283	285	287	289	291	293	295	297	299
301	303	305	307	309	311	313	315	317	319
321	323	325	327	329	331	333	335	337	339
341	343	345	347	349	351	353	355	357	359
361	363	365	367	369	371	373	375	377	379
381	383	385	387	389	391	393	395	397	399
401	403	405	407	409	411	413	415	417	419
421	423	425	427	429	431	433	435	437	439
441	443	445	447	449	451	453	455	457	459
461	463	465	467	469	471	473	475	477	479
481	483	485	487	489	491	493	495	497	499

2.7.3 Ecrire un programme en python qui renvoie la liste de tous les nombres premiers inférieurs à 10^6. Utiliser le crible d'Eratosthène.

2.7.4 Trouver un facteur de 4841.

2.7.5 Montrer que 521 est un nombre premier.

2.7.6 Factoriser les nombres ci-dessous.

a) 200
b) 150
c) 128
d) 164
e) 35
f) 77
g) 143
h) 323
i) 713
j) 1147
k) 473
l) 493
m) 4897
n) 6319
o) 1591
p) 901

2.8 RSA

2.8.1 Chiffrer et déchiffrer le message $m = 8$ à l'aide du système RSA, avec $p = 7$, $q = 11$ et $e = 17$.

2.8.2 Connaissant la clef publique (n, e) d'un individu, déterminer la clé privée d dans les cas suivants :

a) $n = 1073$, $e = 117$;
b) $n = 1073$, $e = 115$;
c) $n = 111$, $e = 31$;
d) $n = 117$, $e = 17$;
e) $n = 105$, $e = 13$;
f) $n = 10001$, $e = 187$.

2.8.3 Un ennemi intercepte le message chiffré $c = 10$, dont le destinataire possède la clef publique $e = 5$, $n = 35$. Quel est le texte clair m ?

2.8.4 Soit $p = 23$ et $q = 31$. On donne encore $e = 17$.

a) Calculer d, l'inverse de e modulo $(p-1)\cdot(q-1) = 22 \cdot 30 = 660$.
b) Ecrire la clef privée correspondante.
c) Ecrire la clef publique correspondante.
d) Chiffrer le « message » $m = 333$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.
e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

f) Chiffrer $m = 555$.
g) Déchiffrer $c = 100$.

2.8.5 Soit $p = 13$ et $q = 19$. On donne encore $e = 11$.

a) Calculer d, l'inverse de e modulo $(p-1)\cdot(q-1)$.
b) Ecrire la clef privée correspondante.

c) Ecrire la clef publique correspondante.

d) Chiffrer le « message » $m = 123$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.

e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

f) Chiffrer $m = 222$.

g) Déchiffrer $c = 55$.

2.8.6 Soit $p = 7$ et $q = 23$. On donne encore $e = 5$.

a) Calculer d, l'inverse de e modulo $(p-1) \cdot (q-1)$.

b) Ecrire la clef privée correspondante.

c) Ecrire la clef publique correspondante.

d) Chiffrer le « message » $m = 111$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.

e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

f) Chiffrer $m = 22$. Commenter le résultat.

g) Déchiffrer $c = 100$.

2.8.7 Un professeur envoie ses notes au secrétariat de l'école par mail. La clef publique du professeur est (3; 55) et celle du secrétariat est (3; 33).

a) Vérifier que la clef du professeur (supposée connue de lui seul) est 27 et que celle du secrétariat est 7.

b) Pour assurer la confidentialité de ses messages, le professeur chiffre les notes avec la clef RSA du secrétariat. Quel est le message chiffré qui correspond à la note 12 ?

c) Pour assurer l'authenticité des messages contenant les notes, le professeur signe ses messages pour le secrétariat après les avoir chiffrés. Le secrétariat reçoit le message 23. Quelle est la note correspondante ?

2.8.8 Définir les termes cryptographie, cryptanalyse et cryptologie.

2.8.9 Bob utilise le protocole RSA et publie sa clef publique $N = 187$ et $e = 3$.

a) Encoder le message $m = 15$ avec la clef publique de Bob.

b) En utilisant le fait que $\varphi(N) = 160$, retrouver la factorisation de N, puis la clef privee de Bob.

2.8.10 Bob_1 et Bob_2 ont pour clef publique RSA respectivement (N, e_1) et (N, e_2) avec e_1 et e_2 premiers entre eux.
Alice envoie le même message m crypté avec les clefs publiques RSA de Bob_1 et Bob_2 en c_1 et c_2.
Expliquer comment Eve, qui intercepte les deux messages cryptés et qui connait les clefs publiques de Bob_1 et Bob_2, peut retrouver le message clair m.

2.8.11 Supposons que l'entier n soit le produit de deux nombres premiers p et q proches (on peut toujours supposer que $p > q$).
On pose $t = \dfrac{p+q}{2}$ et $s = \dfrac{p-q}{2}$.
Montrer que :

a) l'entier s est petit ;

b) $n = t^2 - s^2$;

c) t est légèrement supérieur à la racine carrée de n ;

d) à l'aide de ces informations, écrire un programme pour factoriser n ;

e) appliquer cet algorithme pour factoriser 899, 110417 et 364957402 ;

f) Trouver la clef secrète d correspondante à ($n = 51983$, $e = 17$).

2.8.12 Le fichier `nombres_p_q_1.txt` contient deux nombres premiers `p` et `q`.
On choisit `e = 65537` qui satisfait la condition $\gcd(e, \varphi(n)) = 1$, où $n = p \cdot q$. On dispose ainsi d'une clé publique `(n, e)`.

a) Dans un fichier `clef.py`, construire la clé privée `(n, d)`. Afficher à la console la valeur de `d`.

b) Coder le message suivant (présent dans `message_1.txt`) à l'aide de la clé publique `(n, e)` et l'afficher à la console :

```
12121212121212121212121212121212121212121212121212121212121212
```

c) Décoder le message chiffré suivant (présent dans `chiffre_1.txt`) à l'aide de la clé privée `(n, d)` et l'afficher à la console.

```
25567439145662233958692485794698610817081042607044
76375740884794753357993630922231929342522549252625
83666181770268506210137463778407056604376942495443
54344442167885488175571510178529674397470719290002
50564288716583429743123017673776013168016434939838
37662442086433370446824338338022627966782937829223
36951763006508399321713627145581271533380725975341
78087248990732332457837418659413481406361983817252
46437822132392448183737304695659768807011137853820
```

236969530410283885955315364551645934249457465617893488436619967048244861665350955542794751125486269641398200342295055677990818953963680224811410674869657356236000829

2.9 Solutions des exercices

2.1.1 a) `VVVIXGGTPTMOFVADWST`

b) `INTROUVABLE`

c) `THEOBALD`

2.1.2 Vigenère apporte un plus car une même lettre peut être chiffrée différemment dans le texte ce qui complique l'analyse statistique. Vigenère est donc plus solide.

2.1.3

a) vvvixggtptmofvadwst

b) -

2.1.4 Les sanglots longs des violons de l'automne blessent mon cœur d'une langueur monotone

2.1.5 –

2.1.6 –

2.1.7 –

2.2.1 L'ensemble D des diviseurs communs de a et b n'est pas vide, car $1 \mid a$ et $1 \mid b$. Vu que l'ensemble des diviseurs de a est fini, D est également fini. Vu que D est fini, cet ensemble admet un maximum, noté d. Le nombre d n'est autre que $\gcd(a, b)$.

2.2.2

a) $\{1, 2, 4, 8, 16\}$,

b) $\{1, 3, 5, 15\}$,

c) $\{1\}$.

2.2.3

a) 5

b) 3

c) 1

2.2.4 17

2.2.5 $21331 = 83 \cdot 257$ et $43947 = 3 \cdot 3 \cdot 19 \cdot 257$.
On peut donc écrire $\gcd(21331, 43947) = 257$.

2.2.6 $210632 = 2 \cdot 2 \cdot 2 \cdot 113 \cdot 233$ et $423137 = 11 \cdot 11 \cdot 13 \cdot 269$.
On peut donc écrire $\gcd(21331, 43947) = 1$.

2.2.7 Soit $m \in \mathbb{N}$ tel que $m \mid n$ et $m \mid (n+1)$. Le nombre m doit alors diviser la différence $(n+1) - n$. Ce qui fait que $m \mid 1$. Cela implique finalement que $m = 1$. On a donc $\gcd(n, n+1) = 1$.

2.2.8 Supposons que $a \mid b$. Dans ce cas, a est un diviseur commun de a et b. C'est forcément le plus grand, car si $d \mid a$, alors $d \leq a$.

2.2.9 Notons $d = \gcd(a, b)$. On sait que $d \mid a$ et $d \mid b$, ce qui fait que $d \mid a \cdot r$ et $d \mid b \cdot r$. Par conséquent, $d \mid (a \cdot r + b \cdot s)$. Finalement, $d \mid 1$ et donc $d = 1$.

2.3.1 –

2.3.2 –

2.3.3 –

2.3.4 –

2.3.5 —

2.3.6 —

2.3.7 —

2.3.8 —

2.3.9 —

2.4.1 —

2.4.2 —

2.4.3 —

2.5.1 —

2.5.2 –

2.5.3

```
def beo_bin(n):
    while n:
        print(n%2)
        n = n//2
beo_bin(127)
```

2.5.4

a) $25^5 \mod 133$

```
25^1 mod 133 = 25
25^2 mod 133 = 93
25^4 mod 133 = 4
25 * 4 mod 133 = 100
```

b) $100^{65} \mod 133$

```
100^1 mod 133 = 100
100^2 mod 133 = 25
100^4 mod 133 = 93
100^8 mod 133 = 4
100^16 mod 133 = 16
100^32 mod 133 = 123
100^64 mod 133 = 100
100 * 100 mod 133 = 25
```

c) $107^5 \mod 133$

```
107^1 mod 133 = 107
107^2 mod 133 = 11
107^4 mod 133 = 121
107 * 121 mod 133 = 46
```

d) $46^{65} \mod 133$

```
46^1 mod 133 = 46
46^2 mod 133 = 121
46^4 mod 133 = 11
46^8 mod 133 = 121
46^16 mod 133 = 11
46^32 mod 133 = 121
46^64 mod 133 = 11
46 * 11 mod 133 = 107
```

e) $234^{65} \mod 667$

```
234^1 mod 667 = 234
234^2 mod 667 = 62
234^4 mod 667 = 509
234^8 mod 667 = 285
234^16 mod 667 = 518
234^32 mod 667 = 190
234^64 mod 667 = 82
234 * 82 mod 667 = 512
```

f) $447^{65} \mod 667$

```
447^1 mod 667 = 447
447^2 mod 667 = 376
447^4 mod 667 = 639
447^8 mod 667 = 117
447^16 mod 667 = 349
```

```
447^32 mod 667 = 407
447^64 mod 667 = 233
447 * 233 mod 667 = 99
```

g) $99^{417} \mod 667$

```
99^1 mod 667 = 99
99^2 mod 667 = 463
99^4 mod 667 = 262
99^8 mod 667 = 610
99^16 mod 667 = 581
99^32 mod 667 = 59
99^64 mod 667 = 146
99^128 mod 667 = 639
99^256 mod 667 = 117
99 * 59 * 639 * 117 mod 667 = 447
```

h) $664^{65} \mod 667$

```
664^1 mod 667 = 664
664^2 mod 667 = 9
664^4 mod 667 = 81
664^8 mod 667 = 558
664^16 mod 667 = 542
664^32 mod 667 = 284
664^64 mod 667 = 616
664 * 616 mod 667 = 153
```

2.5.5 —

2.5.6 —

2.6.1 –

2.6.2 –

2.6.3 –

2.6.4 –

2.6.5 –

2.6.6 –

2.6.7 Pour tout $1 \leq a \leq 6$, on a $a^6 \equiv 1 \mod 7$.

2.6.8 Pour tout $1 \leq a \leq 16$, on a $a^{16} \equiv 1 \mod 17$.

2.6.9 On peut émettre l'hypothèse que si p est premier, alors $a^p \equiv a \mod p$. C'est un théorème, appelé le « petit théorème de Fermat ».

2.6.10 Soit $(1, 2, 4, 7, 8, 11, 13, 14)$ la liste des nombres inversibles de $\mathbb{Z}_{15}$ et a un nombre de cette liste. Il a 8 nombres inversibles. On construit une nouvelle liste comme suit :

$$\ell = (a \cdot 1, a \cdot 2, a \cdot 4, a \cdot 7, a \cdot 8, a \cdot 11, a \cdot 13, a \cdot 14)$$

Voyons pourquoi cette liste ne compte que des éléments distincts : Soit $a \cdot z_1$ et $a \cdot z_2$ deux éléments de ℓ et supposons que

$$a \cdot z_1 \mod 15 = a \cdot z_2 \mod 15$$

Notons encore a' l'inverse de a modulo 15. On a alors

$$a' \cdot a \cdot z_1 \mod 15 = a' \cdot a \cdot z_2 \mod 15$$

ce qui veut dire que

$$z_1 \mod 15 = z_2 \mod 15$$

vu que $a' \cdot a \mod 15 = 1$. Cela signifie que si $z_1 \neq z_2$, alors $a \cdot z_1 \neq a \cdot z_2$, modulo 15 bien entendu.
La liste ℓ compte donc 8 éléments, tous inversibles, vu que l'inverse de $a \cdot z$ est donné par $z' \cdot a'$ si z' désigne l'inverse de z modulo 15. On peut donc dire que ℓ est la liste des inversibles de $\mathbb{Z}_{15}$, éventuellement placés dans un ordre différent.

Cela signifie que

$$1 \cdot 2 \cdot 4 \cdot 7 \cdot 8 \cdot 11 \cdot 13 \cdot 14 \equiv a \cdot 1 \cdot a \cdot 2 \cdot a \cdot 4 \cdot a \cdot 7 \cdot a \cdot 8 \cdot a \cdot 11 \cdot a \cdot 13 \cdot a \cdot 14 \mod 15$$

et donc que

$$1 \cdot 2 \cdot 4 \cdot 7 \cdot 8 \cdot 11 \cdot 13 \cdot 14 \equiv 1 \cdot 2 \cdot 4 \cdot 7 \cdot 8 \cdot 11 \cdot 13 \cdot 14 \cdot a \cdot a \cdot a \cdot a \cdot a \cdot a \cdot a \cdot a \mod 15$$

Après simplification, on obtient $1 \equiv a^8 \mod 15$. La simplification est possible vu que tous les nombres impliqués dans l'écriture ci-dessus sont des inversibles.

2.6.11 Il y a 12 inversibles dans $\mathbb{Z}_{21}$. Un argument analogue à celui donné à l'exercice 2.6.10 donne la congruence $a^{12} \equiv 1 \mod 21$ si a est inversible.

2.6.12 –

2.6.13 —

2.6.14 —

2.6.15 —

2.6.16 —

2.6.17 —

2.6.18 —

2.6.19 —

2.7.1 Soit n un nombre entier composé. Cela signifie que $n = a \cdot b$ avec a et b des entiers différents de 1. Supposons que $a > \sqrt{n}$ et que $b > \sqrt{n}$. On peut, dans ce cas, écrire que $a \cdot b > \sqrt{n} \cdot \sqrt{n} = n$, ce qui est contradictoire. L'un des deux facteurs est donc plus petit ou égal à $\sqrt{n}$.

2.7.2

```
  2    3    5    7   11   13   17   19   23   29
 31   37   41   43   47   53   59   61   67   71
 73   79   83   89   97  101  103  107  109  113
127  131  137  139  149  151  157  163  167  173
179  181  191  193  197  199  211  223  227  229
233  239  241  251  257  263  269  271  277  281
283  293  307  311  313  317  331  337  347  349
353  359  367  373  379  383  389  397  401  409
419  421  431  433  439  443  449  457  461  463
467  479  487  491  499
```

2.7.3

```
def listePremiers(borne):
    liste_premiers = list(range(3, borne, 2))
    liste_premiers = [2] + liste_premiers
    r = int(borne**(1/2))

    n = 3
    j = 1
    t = len(liste_premiers)

    while n <= r:
        for i in range(j + n, t, n):
            if liste_premiers[i]:
                liste_premiers[i] = 0
        j += 1
        while not liste_premiers[j]:
            j += 1
        n = liste_premiers[j]

    premiers = []
```

```
    for p in liste_premiers:
        if p:
            premiers.append(p)
    return premiers
```

2.7.4 $4841 = 103 \cdot 47$

2.7.5 Vu que $\sqrt{521} \simeq 22.83$, si 521 n'admet pas de facteur premier inférieur à 22, c'est un nombre premier. Aucun des nombres de la liste

$$(2, 3, 5, 7, 11, 13, 17, 19)$$

n'étant un diviseur de 521, c'est un nombre premier.

2.7.6 On donne ci-dessous la liste des facteurs premiers de chaque nombre :

a) $(2, 2, 2, 5, 5)$
b) $(2, 3, 5, 5)$
c) $(2, 2, 2, 2, 2, 2, 2)$
d) $(2, 2, 41)$
e) $(5, 7)$
f) $(7, 11)$
g) $(11, 13)$
h) $(17, 19)$
i) $(23, 31)$
j) $(31, 37)$
k) $(11, 43)$
l) $(17, 29)$
m) $(59, 83)$
n) $(71, 89)$
o) $(37, 43)$
p) $(17, 53)$

2.8.1 —

2.8.2 —

2.8.3 $m = 5$

2.8.4 Soit $p = 23$ et $q = 31$. On donne encore $e = 17$.

a) Calculer d, l'inverse de e modulo $(p-1) \cdot (q-1) = 22 \cdot 30 = 660$.

```
660 = 1 * 660 + (0) * 17
17 = 0 * 660 + (1) * 17
14 = 1 * 660 + (-38) * 17
3 = -1 * 660 + (39) * 17
2 = 5 * 660 + (-194) * 17
1 = -6 * 660 + (233) * 17
```

b) Ecrire la clef privée correspondante :

$$(23; 31; 233)$$

c) Ecrire la clef publique correspondante :

$$(713; 17)$$

d) Chiffrer le « message » $m = 333$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.

```
333^1 = 333
333^2 = 374
333^4 = 128
333^8 = 698
333^16 = 225
333 * 225 = 60
```

e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

```
60^1 = 60
60^2 = 35
60^4 = 512
60^8 = 473
60^16 = 560
60^32 = 593
60^64 = 140
60^128 = 349
60 * 473 * 593 * 140 * 349 = 333
```

f) Chiffrer $m = 555$.

```
555^1 = 555
555^2 = 9
555^4 = 81
555^8 = 144
555^16 = 59
555 * 59 = 660
```

g) Déchiffrer $c = 100$.

```
100^1 = 100
100^2 = 18
100^4 = 324
100^8 = 165
100^16 = 131
100^32 = 49
100^64 = 262
100^128 = 196
100 * 165 * 49 * 262 * 196 = 41
```

2.8.5 Soit $p = 13$ et $q = 19$. On donne encore $e = 11$.

a) Calculer d, l'inverse de e modulo $(p-1)\cdot(q-1) = 12 \cdot 18 = 216$.

```
216 = 1 * 216 + (0) * 11
11 = 0 * 216 + (1) * 11
7 = 1 * 216 + (-19) * 11
4 = -1 * 216 + (20) * 11
3 = 2 * 216 + (-39) * 11
1 = -3 * 216 + (59) * 11
```

b) Ecrire la clef privée correspondante :

$$(13; 19; 59)$$

c) Ecrire la clef publique correspondante :

$$(247; 11)$$

d) Chiffrer le « message » $m = 123$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.

```
123^1 = 123
123^2 = 62
123^4 = 139
123^8 = 55
123 * 62 * 55 = 24
```

e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

```
24^1 = 24
24^2 = 82
24^4 = 55
24^8 = 61
24^16 = 16
24^32 = 9
24 * 82 * 61 * 16 * 9 = 123
```

f) Chiffrer $m = 222$.

```
222^1 = 222
222^2 = 131
222^4 = 118
222^8 = 92
222 * 131 * 92 = 40
```

g) Déchiffrer $c = 55$.

```
55^1 = 55
55^2 = 61
```

```
55^4 = 16
55^8 = 9
55^16 = 81
55^32 = 139
55 * 61 * 9 * 81 * 139 = 139
```

2.8.6 Soit $p = 7$ et $q = 23$. On donne encore $e = 5$.

a) Calculer d, l'inverse de e modulo $(p-1)\cdot(q-1) = 6 \cdot 22 = 132$.

```
132 = 1 * 132 + (0) * 5
5 = 0 * 132 + (1) * 5
2 = 1 * 132 + (-26) * 5
1 = -2 * 132 + (53) * 5
```

b) Ecrire la clef privée correspondante :

$$(7; 23; 53)$$

c) Ecrire la clef publique correspondante :

$$(161; 5)$$

d) Chiffrer le « message » $m = 111$ en utilisant la formule

$$c = m^e \mod n$$

si $n = p \cdot q$.

```
111^1 = 111
111^2 = 85
111^4 = 141
111 * 141 = 34
```

e) Retrouver le message à partir du chiffre c en utilisant la formule

$$m = c^d \mod n$$

```
34^1 = 34
34^2 = 29
34^4 = 36
34^8 = 8
34^16 = 64
34^32 = 71
34 * 36 * 64 * 71 = 111
```

f) Chiffrer $m = 22$.

```
22^1 = 22
22^2 = 1
22^4 = 1
22 * 1 = 22
```

Le chiffre est identique au message. C'est une situation critique, à éviter.

g) Déchiffrer $c = 100$.

```
100^1 = 100
100^2 = 18
100^4 = 2
100^8 = 4
100^16 = 16
100^32 = 95
100 * 2 * 16 * 95 = 32
```

2.8.7

a) Pour le professeur $\varphi(55) = 40$ et $27 \cdot 3 = 81 = 1 \pmod{40}$.

Pour le secrétariat $\varphi(33) = 20$ et $7 \cdot 3 = 21 = 1 \pmod{20}$.

b) Le professeur envoie $m = 12 \pmod{33}$. Or $12^2 = 12 \pmod{33}$; donc $m = 12 \pmod{33}$.

c) La note $(23^3 \pmod{55})^7 \pmod{33} \equiv 12^7 \pmod{33} \equiv 12$.

2.8.8 La cryptographie est l'art de rendre inintelligible, de crypter, de coder, un message pour ceux qui ne sont pas habilités à en prendre connaissance. Le chiffre, le code est le procédé, l'algorithme, la fonction, qui permet de crypter un message.
La cryptanalyse est l'art pour une personne non habilitée, de décrypter, de décoder, de déchiffrer, un message. C'est donc l'ensemble des procédés d'attaque d'un système cryptographique.
La cryptologie est l'ensemble formé de la cryptographie et de la cryptanalyse.

2.8.9

a) Le message codé est $c \equiv 15^3 \pmod{187} \equiv 9$.

b) Ecrivons $N = p\,q$. On a donc $\varphi(N) = (p-1)(q-1) = pq-p-q+1 = N-(p+q)+1$, et ainsi $p + q = N - \varphi(N) + 1 = 187 - 160 + 1 = 28$.

Les nombres p et q sont racines du polynôme

$$X^2 - (p+q)X + pq = X - 28\,X + 187$$

Le discriminant est $282-4\cdot 187 = 36$ et ainsi $p = (28-6)/2 = 11$ et $q = (28+6)/2 = 17$.

2.8.10 Puisque e_1 et e_2 sont premiers entre eux, il existe deux entiers u et v tels que $u \cdot e_1 + v \cdot e_2 = 1$. Eve peut calculer u et v, et finalement retrouve le message en faisant

$$c_1{}^u \cdot c_2{}^v \equiv m^{u\,e_1} \cdot m^{v\,e_2} \equiv m^{u\,e_1+v\,e_2} \equiv m \pmod{N}$$

2.8.11

```
# exemple
p=1093
q=1091
n=p*q

# algorithme
from math import sqrt
```

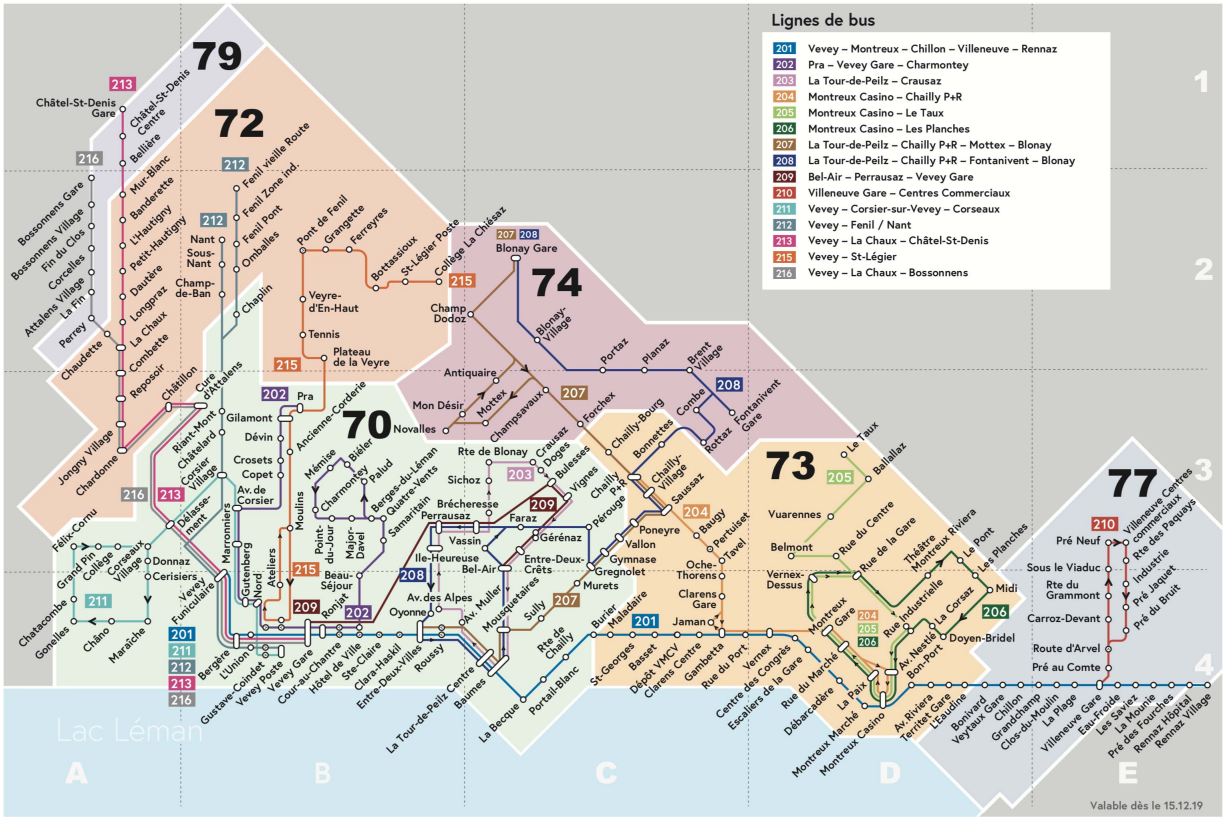
```
t=int(sqrt(n))
z=2
while int(sqrt(z)) != z:
    t += 1
    z = t**2-n

print('p=',t+sqrt(z))
```

Chapitre 3

Graphes

3.1 Généralités

3.1.1 Voici le plan des bus de notre région.



a) Ce plan est-il un graphe ?

b) Que représentent les sommets ?

c) Que représentent les arêtes ?

d) Ce graphe est-il connexe ?

e) Ce graphe est-il simple ou orienté ?

f) Quel est le degré du sommet `Gilamont` ? du sommet `Vevey Gare` ?

3.1.2 Dessiner un graphe représentant les amitiés suivantes parmi quatre personnes :

— John est ami avec Joan and Jill, mais pas avec Jack ;

— Jack est ami avec Jill, mais pas avec Joan ;

— Joan est amie avec Jill.

3.1.3 Dessiner le graphe suivant : les sommets sont les faces d'un cube, deux sommets sont reliés si les faces correspondantes ont une arête du cube en commun.

3.1.4 On a six wagons à trier. Dans la gare de triage, les wagons entrent dans l'ordre 2, 5, 3, 6, 1, 4 et doivent sortir dans l'ordre croissant.

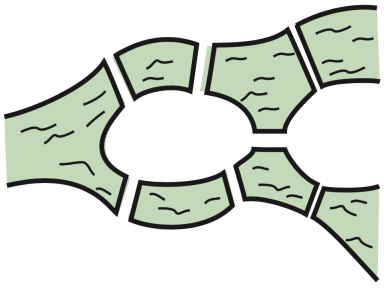
Deux wagons `i` et `j` peuvent être mis sur la même voie si et seulement s'ils entrent dans l'ordre dans lequel ils doivent sortir.

Dessiner un graphe illustrant la situation, en indiquant ce que représentent les sommets et les arêtes de votre graphe.

Quel sera le nombre minimal de voies nécessaires au tri ?

3.1.5 (Les sept ponts de Königsberg)

Au XVIIIe siècle les habitants de Königsberg aimaient se promener le dimanche et traverser les différents ponts de leur ville. Ils se demandaient s'il leur était possible de parcourir la ville en empruntant chacun des 7 ponts une fois et une seule.



Un promeneur veut traverser, une fois et une seule, chacun des sept ponts de la ville.

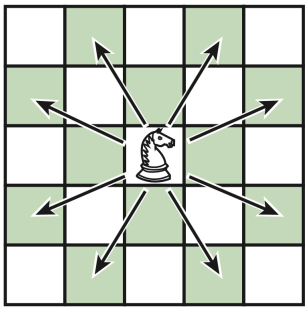
a) Peut-il trouver un itinéraire tel que la région d'arrivée soit la même que celle de départ ?

b) Peut-il trouver un itinéraire tel que les régions d'arrivée et de départ soient distinctes ?

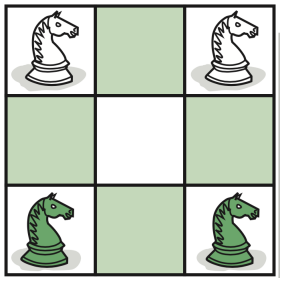
3.1.6 (Cavaliers sur un échiquier 3 x 3)

Dans le jeu d'échecs la pièce dont le déplacement est le plus compliqué est le cavalier. Les possibilités de déplacement d'un cavalier sur un échiquier sont indiquées sur la figure

ci-dessous.



On considère maintenant le mini échiquier 3 x 3 de la figure ci-dessous où sont placés deux cavaliers blancs et deux cavaliers en couleur.



Est-il possible de permuter les deux cavaliers blancs et les deux cavaliers en couleur ?

3.1.7 Démontrer le **théorème des poignées de main** : La somme des degrés des sommets d'un graphe est égale à deux fois le nombre d'arêtes.

3.1.8 Est-il possible de relier 15 ordinateurs de sorte que chaque appareil soit relié avec exactement trois autres ?

3.1.9 Montrer que dans un groupe formé de six personnes, il y en a nécessairement trois qui se connaissent mutuellement ou trois qui ne se connaissent pas (on suppose que si A connaît B, B connaît également A).

Montrer que cela n'est plus nécessairement vrai dans un groupe de cinq personnes.

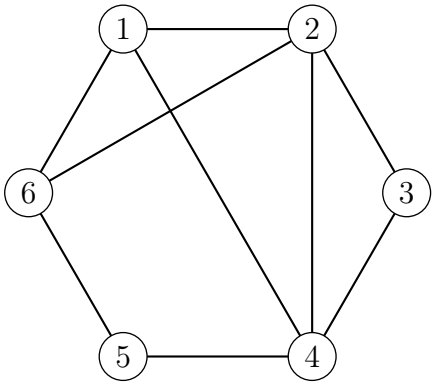
3.1.10

a) Quel est l'ordre du graphe ci-dessous ?

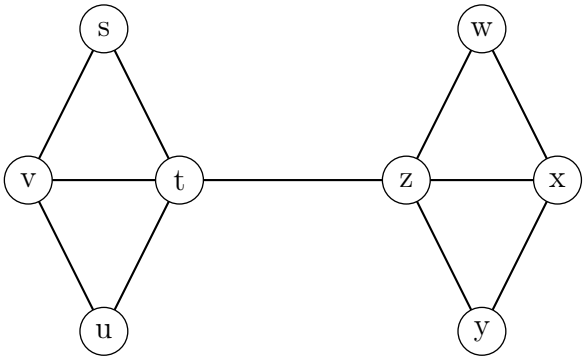
b) Quel est le degré du sommet 1 ? du sommet 4 ?

c) Quels sont les sommets adjacents au sommet 2 ? au sommet 6 ?

d) Il y a deux sommets adjacents chacun à quatre autres sommets. Lesquels ?



3.1.11 Écrire tous les chemins reliant s à y sur le graphe donné ci-dessous. Donner la longueur des chemins trouvés.



Rappel : un chemin est une chaîne telle que chaque arête de celle-ci soit parcourue une et une seule fois.

3.1.12 Dessiner :

a) un graphe connexe avec 8 sommets ;

b) un graphe non connexe avec 8 sommets et deux composantes ;

c) un graphe non connexe avec 8 sommets et trois composantes.

3.1.13

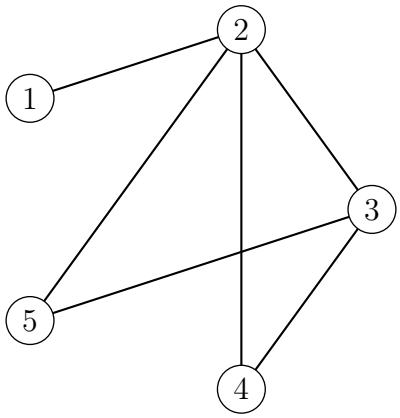
a) Le graphe G ci-dessous est-il complet ?

b) Est-il connexe ?

c) Trouver tous les sous-graphes complets de G (donner la liste de leurs sommets) ?

d) Trouver un chemin de longueur 4 pour aller du sommet 1 au sommet 5 sans passer deux fois par le même sommet.

e) Peut-on trouver un cycle comprenant le sommet 1 ?

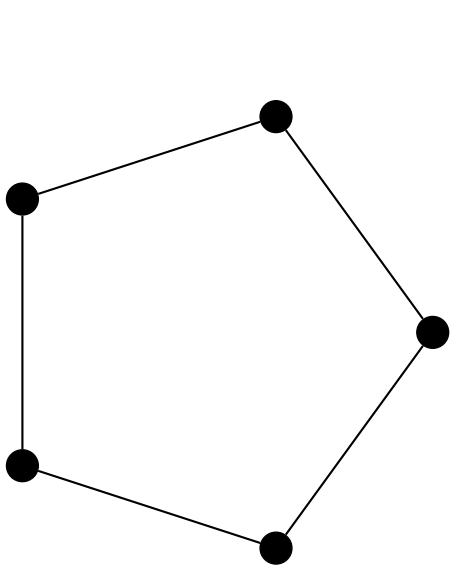


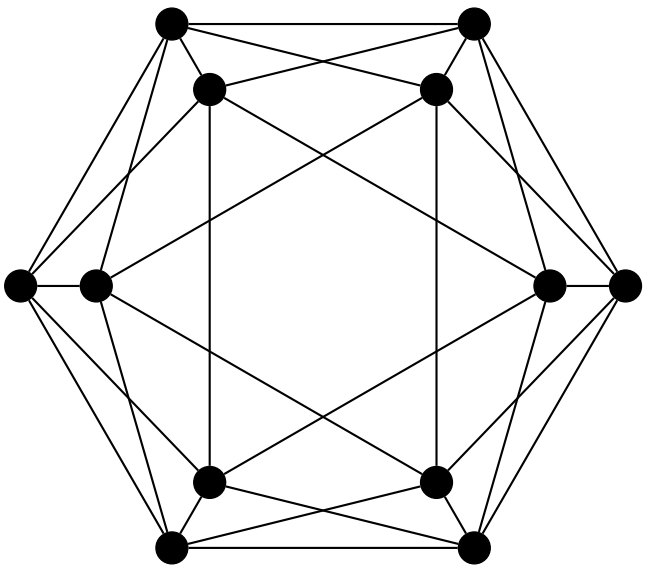
3.1.14 Construire un graphe dont la suite des degrés est (1, 1, 1, 2, 2, 3, 4, 5, 5).

3.1.15 Construire un graphe **connexe** dont la suite des degrés est (1, 1, 2, 3, 3, 4, 4, 6).

3.1.16 Construire un graphe **connexe** dont la suite des degrés est (1, 1, 1, 2, 2, 2, 4, 5, 5).

3.1.17 Soit G un graphe. On dit que G est r-régulier si chaque sommet de G est de degré r. On a le résultat suivant : Si G est r-régulier et qu'il a n sommets, alors G a $n \cdot r/2$ arêtes.
Vérifier ce résultat sur les graphes réguliers suivants :





3.1.18

a) Prouver qu'il n'y a pas de graphe 3-régulier à sept sommets.

b) Prouver que, si n et r sont les deux impairs, il n'y a pas de graphe r-régulier avec n sommets.

3.1.19 Dessiner les 11 graphes simples non étiquetés d'ordre 4.

3.1.20 Dessiner tous les graphes simples non étiquetés d'ordre 5. Il y en a 34.

3.1.21 Notons C_5 le graphe cyclique d'ordre 5 et $\overline{C}_5$ son complémentaire. Montrer que

$$C_5 \simeq \overline{C}_5$$

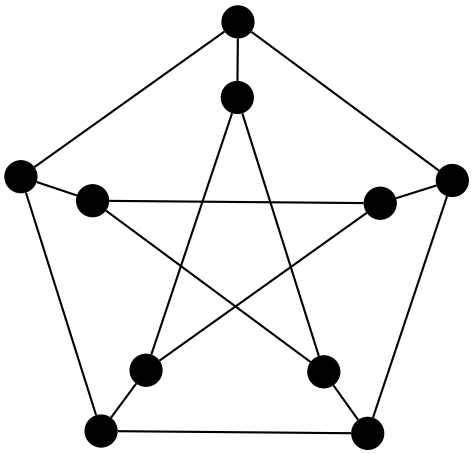
ou, autrement dit, que ces deux graphes sont isomorphes. Montrer ensuite qu'il n'y a pas d'autre graphe cyclique isomorphe à son complémentaire.

3.1.22 Soit G un graphe à ν sommets. Montrer que si $G \simeq \overline{G}$ alors ν ou $\nu - 1$ est un multiple de 4.

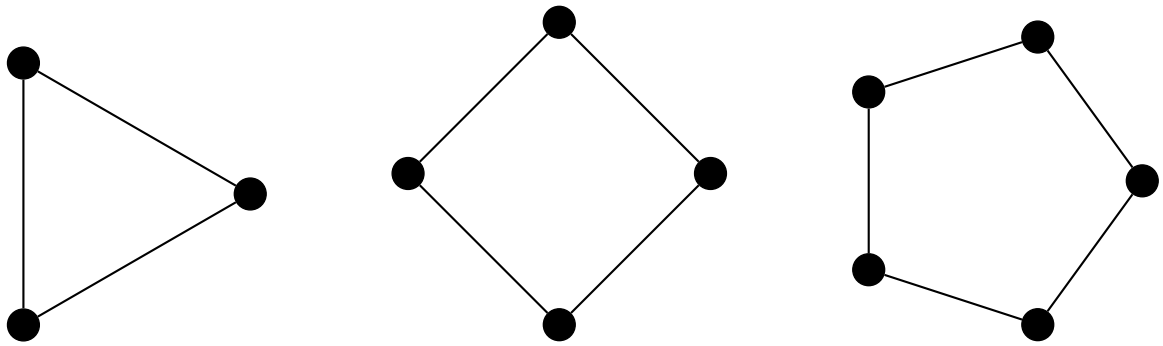
3.1.23 Vu qu'il y a un nombre impair de graphes à quatre sommets, l'un d'entre eux doit être « auto-complémentaire », c'est à dire isomorphe à son complémentaire. Duquel s'agit-il ?

3.2 Graphes eulériens

3.2.1 Soit le graphe de Petersen représenté ci-dessous.

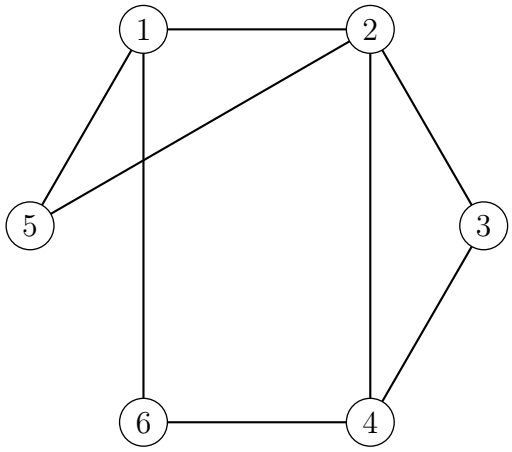


Parmi les graphes représentés ci-dessous, quels sont ceux qui sont un sous-graphe du graphe de Petersen ?



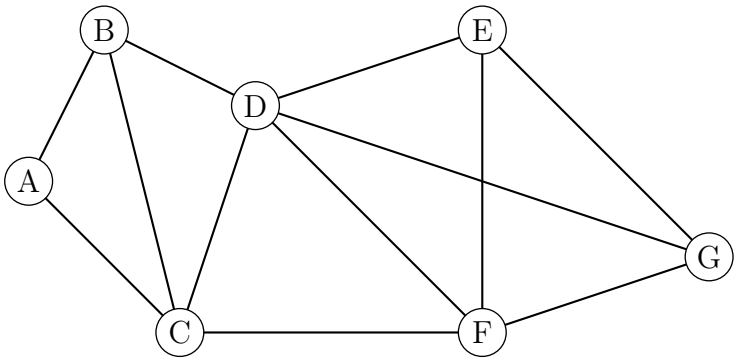
Le graphe de Petersen (1839-1910, mathématicien danois) est, en théorie des graphes, un graphe particulier possédant 10 sommets et 15 arêtes. Il s'agit d'un petit graphe qui sert d'exemple et de contre-exemple pour plusieurs problèmes de la théorie des graphes (Wikipédia).

3.2.2 Soit le graphe ci-dessous.



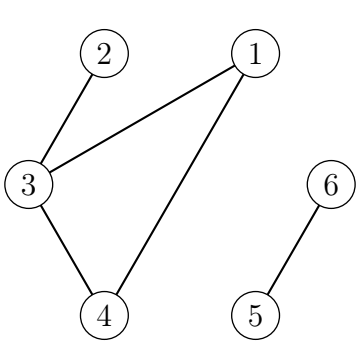
a) Pourquoi le graphe ci-dessus n'admet-il pas de cycle eulérien ?
b) Pourquoi le cycle $1-2-4-3-2-5-1-6$ n'est-il pas une chaîne eulérienne pour ce graphe ?
c) Trouver une chaîne eulérienne d'origine 4.
d) On ajoute une arête de 1 à 4. Montrer qu'il est alors possible de trouver un cycle eulérien.

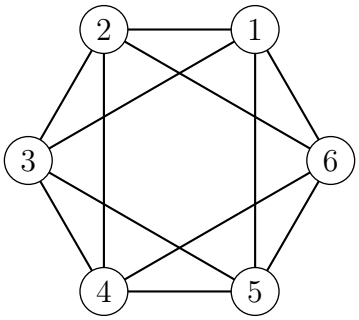
3.2.3 Soit le graphe ci-dessous.

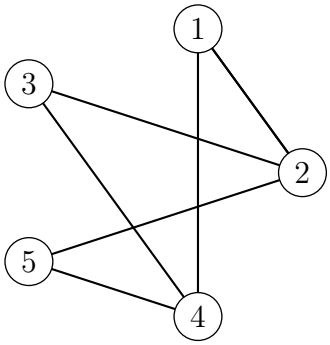


a) Ce graphe admet-il une chaîne eulérienne ? Justifier la réponse. Si oui donner une telle chaîne.
b) Ce graphe admet-il un cycle eulérien ? Justifier la réponse. Si oui donner un tel cycle.

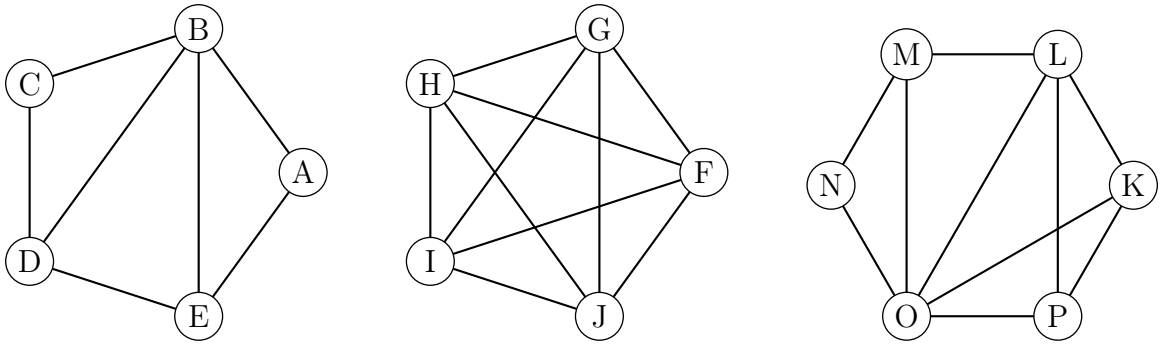
3.2.4 Les graphes ci-dessous sont-ils eulériens (ou semi-eulériens) ?





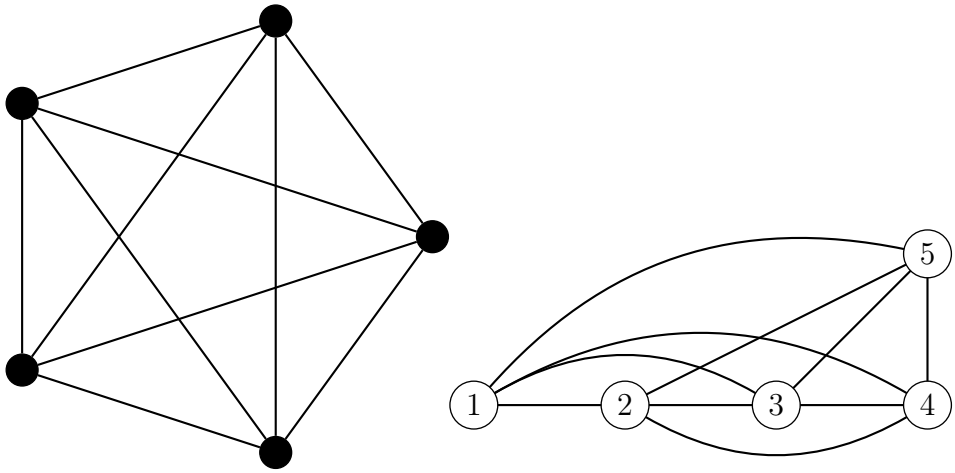


3.2.5 Les graphes ci-dessous admettent-ils un cycle eulérien ou une chaîne eulérienne ?

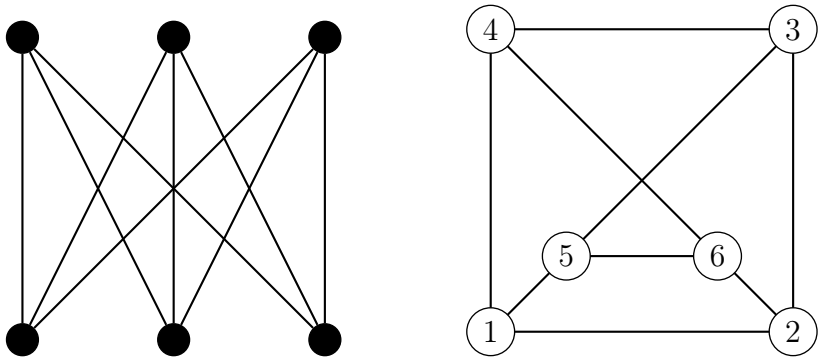


3.2.6 Une chèvre, un chou et un loup se trouvent sur la rive d'un fleuve ; un passeur souhaite les transporter sur l'autre rive mais, sa barque étant trop petite, il ne peut transporter qu'un seul d'entre eux à la fois. Comment doit-il procéder afin de ne jamais laisser ensemble et sans surveillance le loup et la chèvre, ainsi que la chèvre et le chou ?

3.2.7 Montrer que les deux graphes représentés ci-dessous sont isomorphes.



3.2.8 Montrer que les deux graphes représentés ci-dessous sont isomorphes.



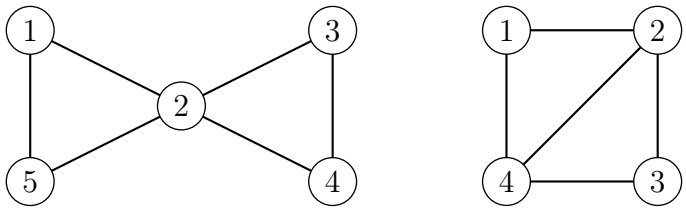
3.3 Arbres

3.3.1 Dessiner tous arbres non étiquetés à 6 sommets ou moins.

3.3.2 En ajoutant à chaque arbre à 6 sommets une arête à la fois, et ceci de toutes les façons possibles, dessiner les 11 arbres à 7 sommets.

3.3.3 En ajoutant à chaque arbre à 7 sommets une arête à la fois, et ceci de toutes les façons possibles, dessiner les 23 arbres à 8 sommets.

3.3.4 On considère les deux graphes ci-dessous :



Pour chaque graphe

a) dessiner tous les arbres couvrants étiquetés ;

b) indiquer ceux qui sont isomorphes.

3.3.5 Trouver tous les arbres couvrants non isomorphes de $K_{3,3}$.

3.3.6 Une *forêt* est un graphe non forcément connexe dont chacune des composantes connexes est un arbre.

a) Soit G une forêt à n sommets et k composantes connexes. Donner le nombre d'arêtes de G.

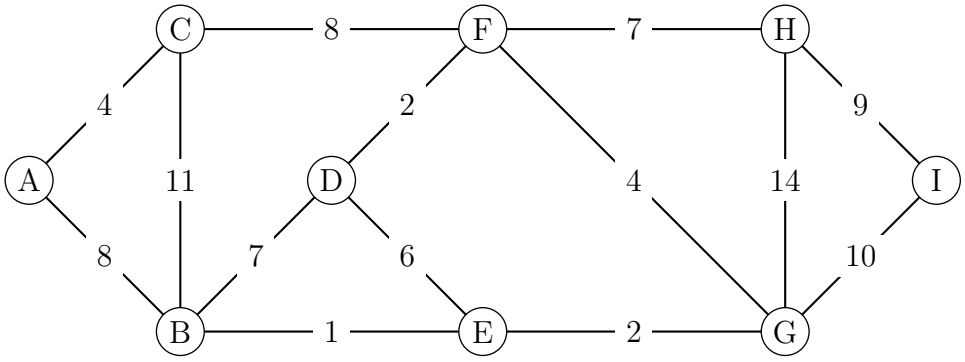
b) Construire une forêt à 12 sommets et 9 arêtes.

c) Est-il vrai que toute forêt à k composantes connexes a au moins $2k$ sommets de degré 1 ?

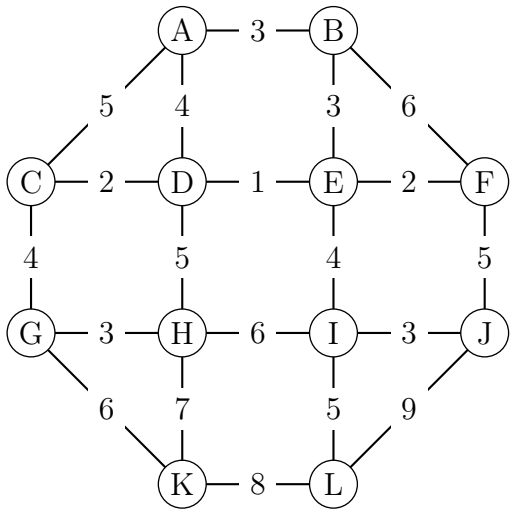
3.3.7 Par l'absurde, démontrer que la suppression d'une arête d'un arbre ne peut le déconnecter en plus de deux composantes connexes.

3.3.8 Par l'absurde, démontrer que l'ajout d'une arête à un arbre ne peut créer plus d'un cycle.

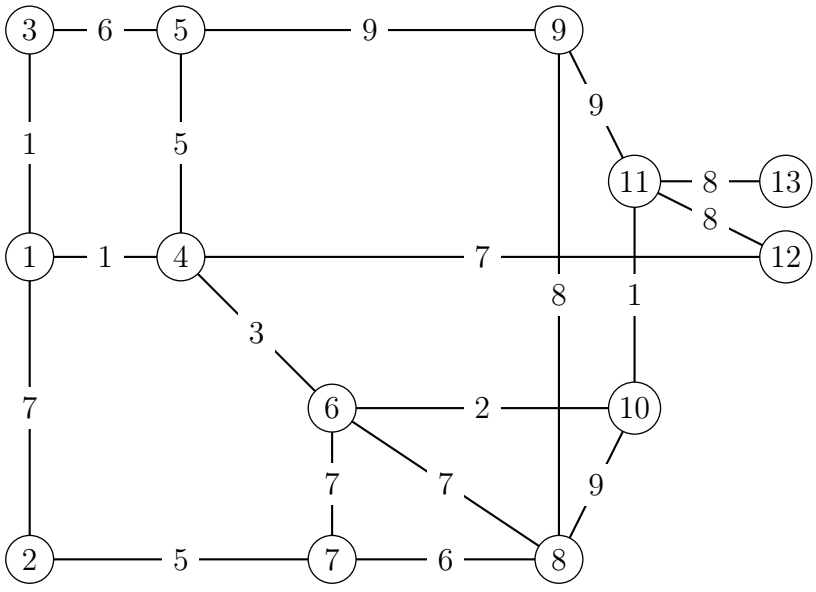
3.3.9 Pour le graphe pondéré ci-dessous, trouver un arbre couvrant de poids minimum.



3.3.10 Pour le graphe pondéré ci-dessous, trouver un arbre couvrant de poids minimum.



3.3.11 Considérons le graphe G ci-dessous.

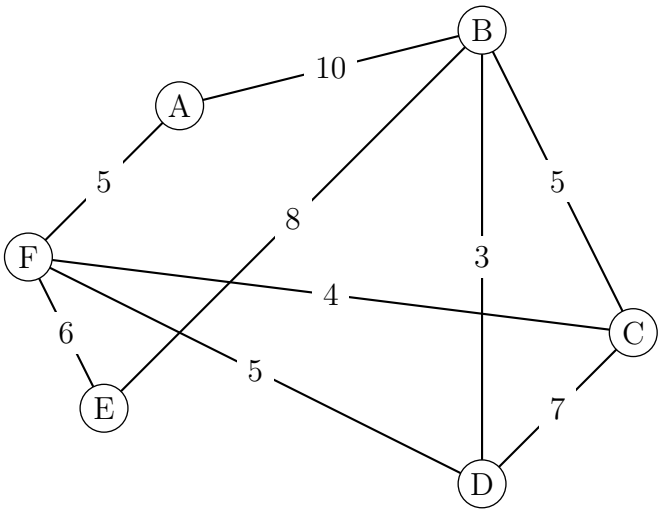


a) Déterminer un arbre de poids minimal de G à l'aide de l'algorithme de Kruskal.

b) Déterminer un arbre de poids minimal de G à l'aide de l'algorithme de Prim, en partant du sommet initial ①.

3.4 Graphes valués : le chemin le plus court

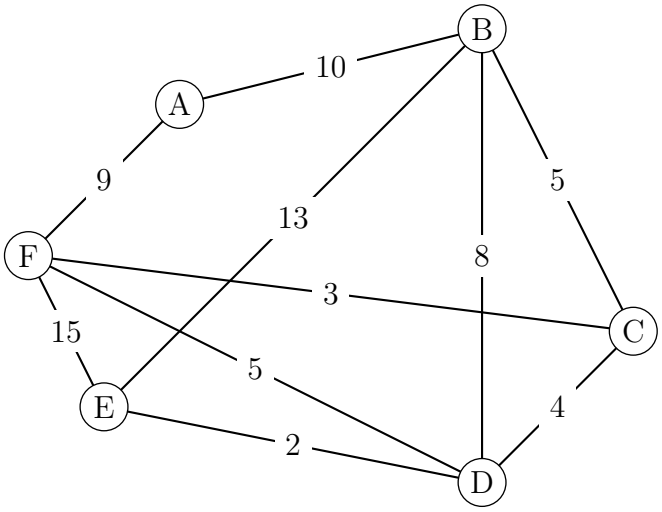
3.4.1 Soit le graphe ci-dessous.



a) Quelle est la longueur du chemin $A - B - C - F - E$?

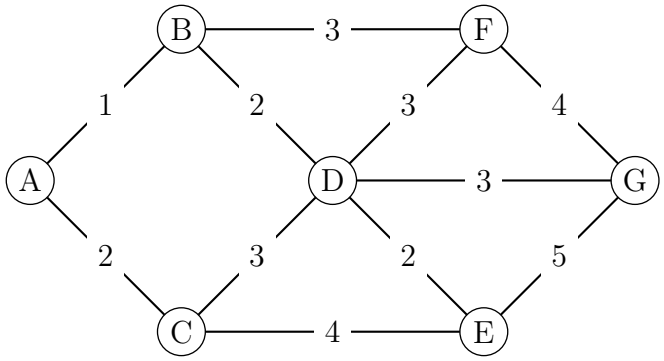
b) Déterminer le chemin de poids minimal reliant F à B.

3.4.2 Soit le graphe ci-dessous.



Déterminer le chemin de poids minimal reliant A à E.

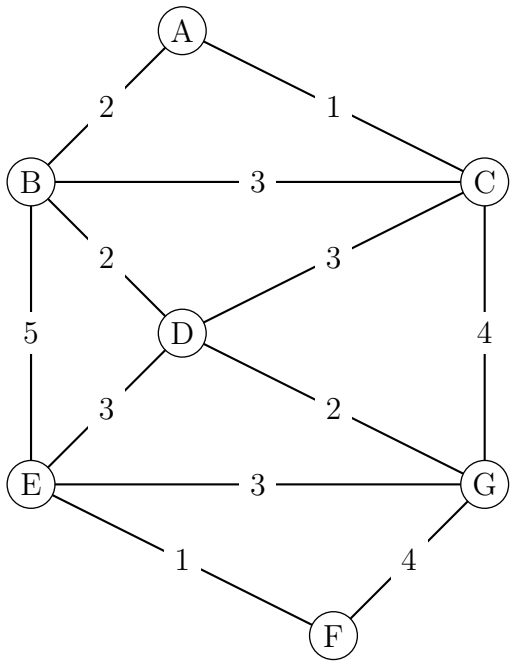
3.4.3 Soit le graphe ci-dessous.



Utiliser l'algorithme de Dijkstra pour déterminer le plus court chemin entre A et G.

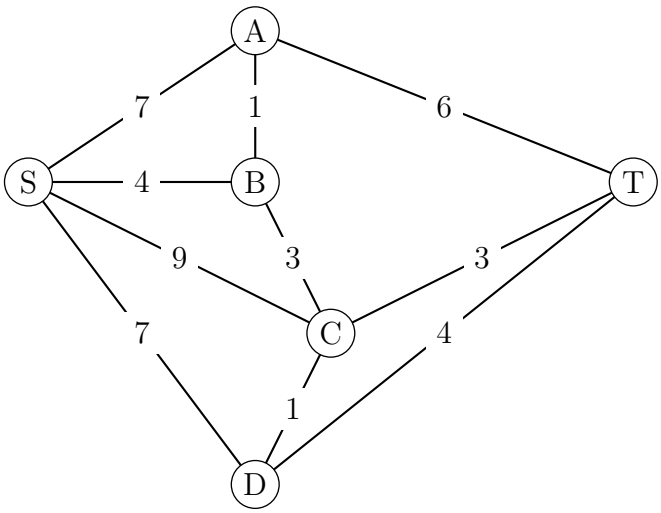
3.4.4 Le graphe présente un réseau routier entre différents points d'une ville.

Chaque tronçon est pondéré par le temps nécessaire, en minute, pour le parcourir.



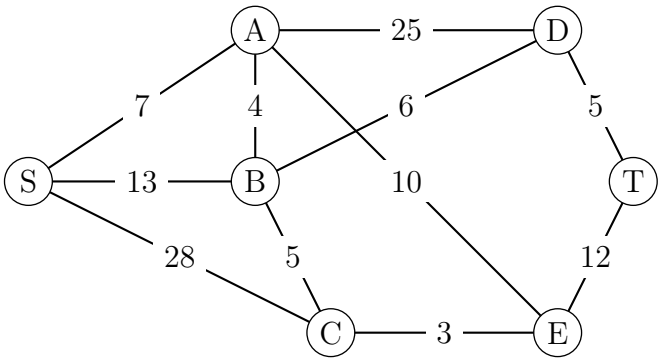
Utiliser l'algorithme de Dijkstra pour déterminer le(s) chemin(s) qui minimise(nt) le temps pour aller de A à F.

3.4.5 Soit le graphe ci-dessous :



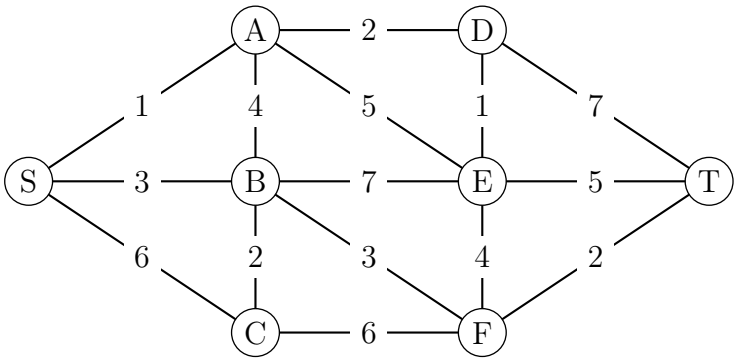
À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de S à T.

3.4.6 Soit le graphe ci-dessous :



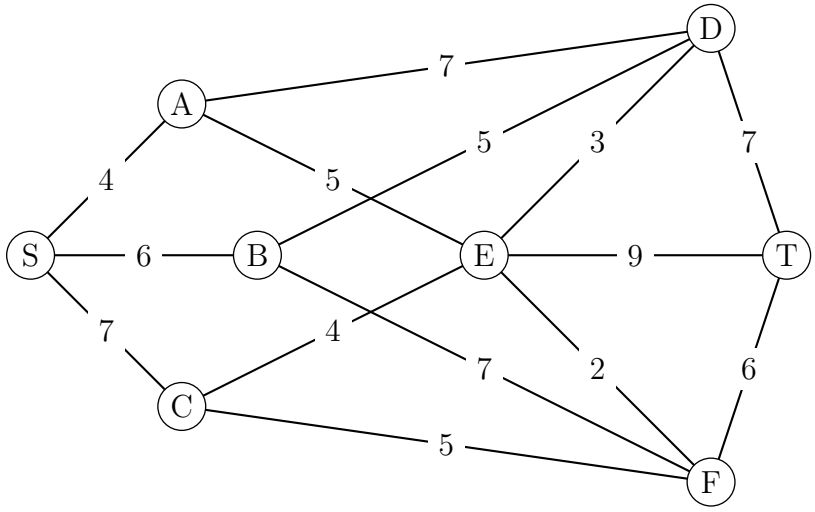
À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de S à T.

3.4.7 Soit le graphe ci-dessous :



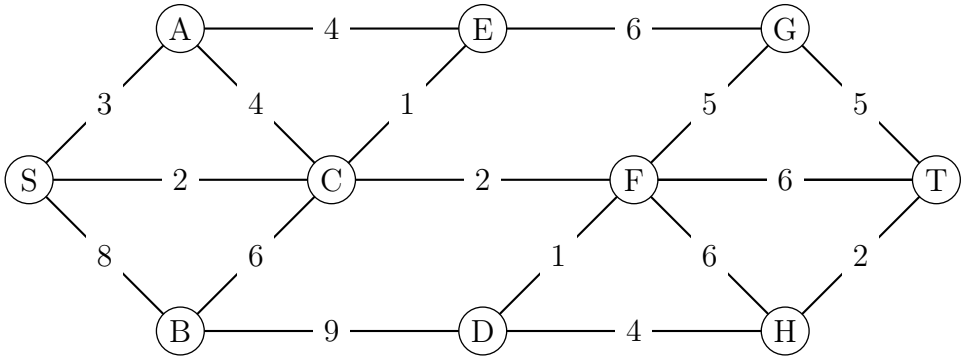
À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de S à T.

3.4.8 Soit le graphe ci-dessous :



À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de S à T.

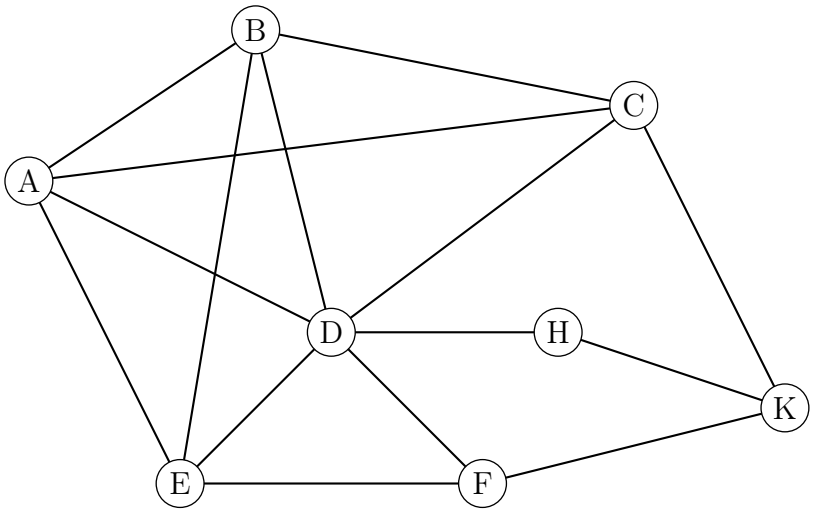
3.4.9 Soit le graphe ci-dessous :



À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de S à T.

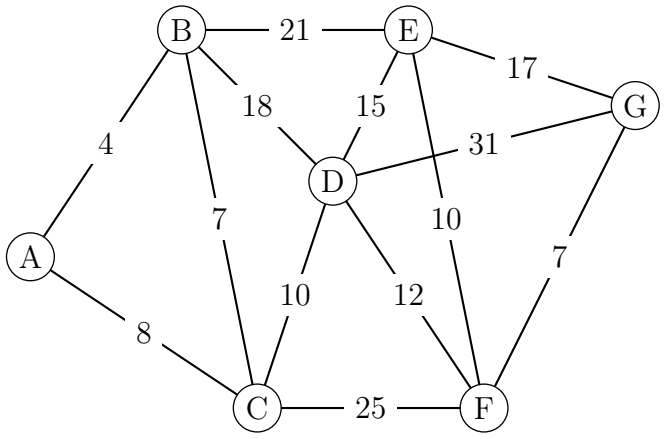
3.4.10 Le graphe ci-dessous représente le plan d'un zoo. Le sommet A désigne son accès, les autres sommets désignent les différents secteurs animaliers de ce zoo.
Une arête représente l'allée reliant deux secteurs et est pondérée par la distance de parcours, exprimée en mètres, entre ces deux secteurs. Les distances sont données dans un tableau.

AB	AC	AD	AE	BC	BD	BE	CD	CK	DE	DF	DH	EF	FK	HK
90	290	175	150	185	155	180	120	260	110	105	220	135	230	145



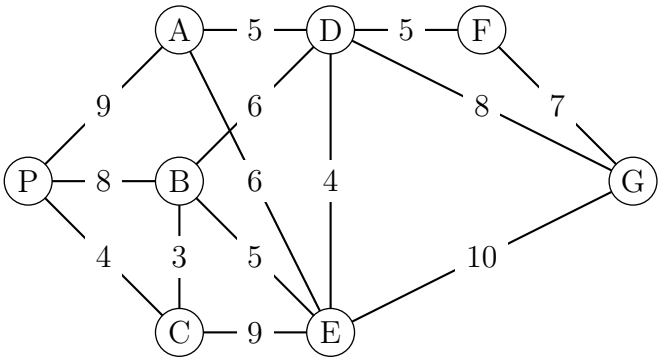
Les services de sécurité basés au sommet A doivent intervenir dans le secteur K. Déterminer, à l'aide de l'algorithme de Dijkstra, l'itinéraire le plus court. Donner sa longueur en mètres.

3.4.11 Une région est munie d'un réseau de train représenté par le graphe ci-dessous.
Les stations sont symbolisées par les sommets A, B, C, D, E, F et G. Chaque arête représente une ligne reliant deux gares. Le temps de parcours en minutes entre chaque sommet ont été rajoutés sur le graphe.



Déterminer, en minutes, le plus court chemin reliant la gare B à la gare G.

3.4.12 Un réseau de navettes gratuites est mis en place entre des parkings situés aux abords de la ville et les principaux sites de la ville.
Le graphe ci-contre indique les voies et les temps des liaisons, en minutes, entre ces différents sites.

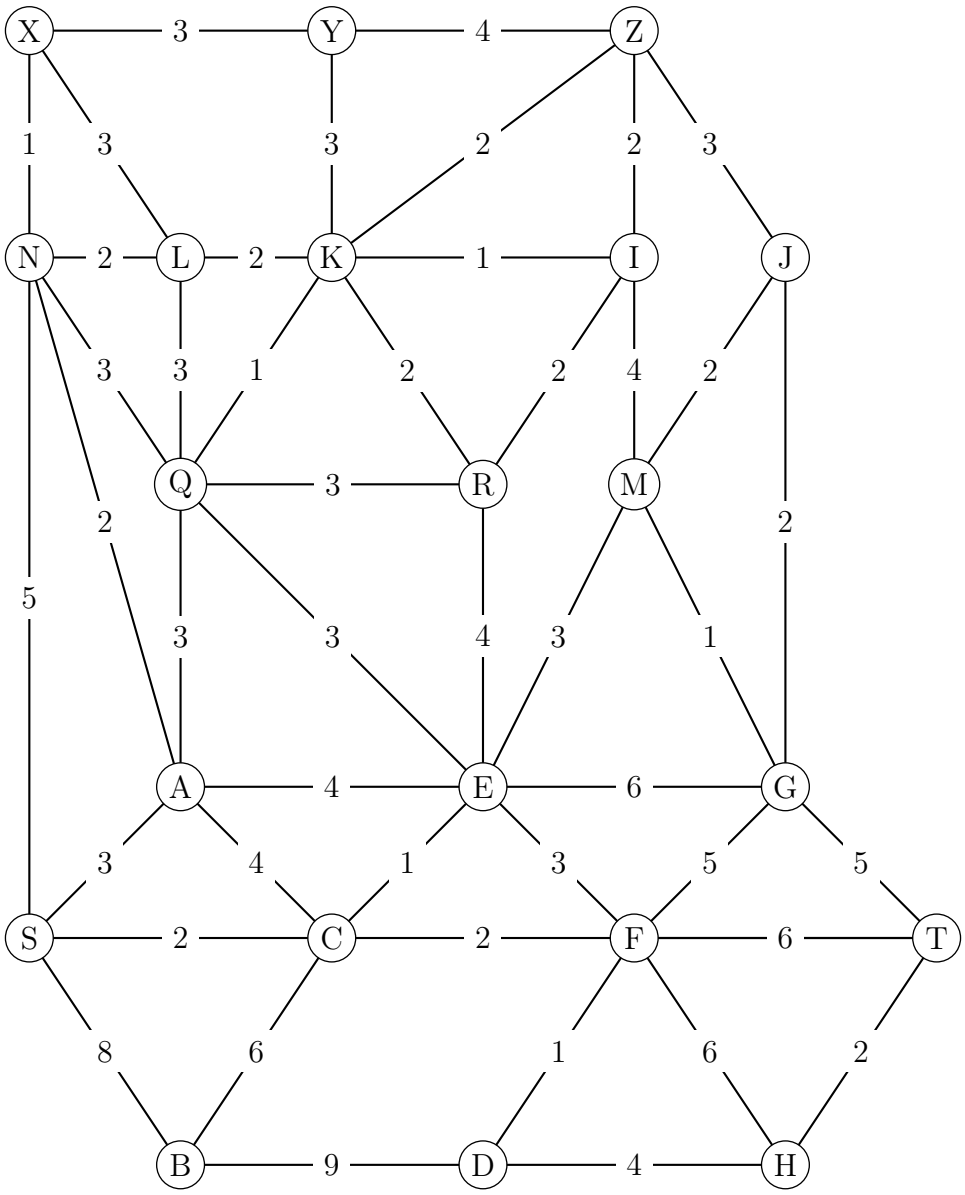


a) Peut-on envisager un itinéraire qui relierait le parking P à la gare G en desservant une et une seule fois tous les sites ?

b) Peut-on envisager un itinéraire qui emprunterait une et une seule fois toutes les voies ?

c) Déterminer un trajet de durée minimale pour se rendre du parking P à la gare G.

3.4.13 Soit le graphe ci-dessous :



À l'aide de l'algorithme de Dijkstra, trouver le chemin le plus court qui mène de X à H.

3.5 Solutions des exercices

3.1.1 a) oui.

b) Un sommet représente une station (ou arrêt) de bus.

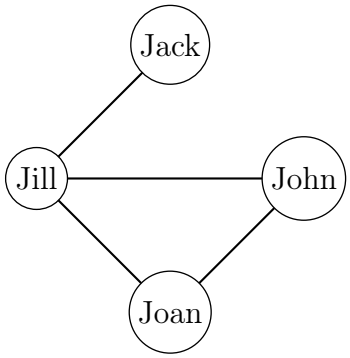
c) Une arêtes représente la route reliant deux stations consécutives.

d) Oui, on peut aller de toute station A (un sommet) à toute station B (un autre sommet).

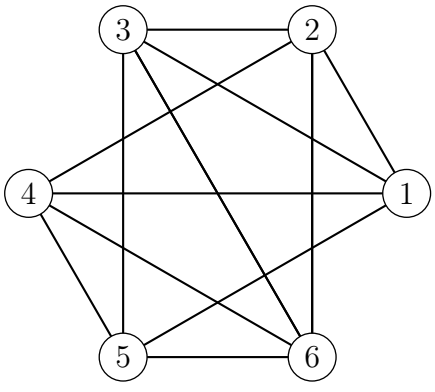
e) Orienté (cf ligne 202 par exemple).

f) 2 et 7.

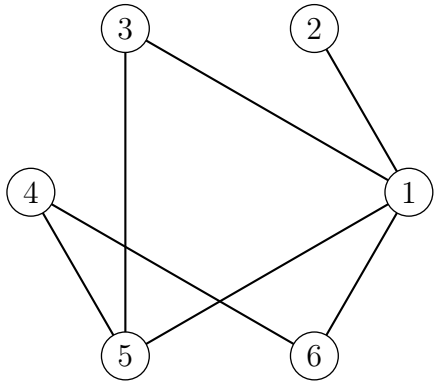
3.1.2



3.1.3



3.1.4 On représente les wagons par les sommets. Une arête relie deux sommets `i` et `j` si les wagons `i` et `j` ne peuvent pas être sur la même voie. On obtient le graphe ci-dessous.



On voit que 1, 3 et 5 ne peuvent pas être sur la même voie. Il faut donc trois voies au minimum.

3.1.5 -

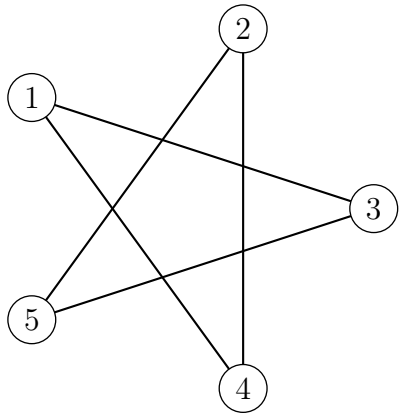
3.1.6 -

3.1.7 Considérons un graphe formé au départ d'un certain nombres de sommets et de 0 arêtes. La somme des degrés est alors 0. Chaque fois que l'on ajoute une arête à ce graphe, la somme des degrés augmente de 2, vu que l'arête relie 2 sommets entre eux. Après l'ajout de n arêtes, la somme des degrés du graphe vaut donc $2\,n$. Vu que tout graphe est obtenu par ce procédé, on obtient bien que la somme des degrés est le double du nombre des arêtes.

3.1.8 Considérons le graphe simple dont les sommets sont les 15 ordinateurs, les arêtes étant les liaisons entre ces ordinateurs. Si chaque appareil est relié à exactement 3 ordinateurs du réseau, les sommets du graphe sont tous de degré impair. D'après un résultat établi, un tel graphe doit posséder un nombre pair de sommets, le réseau ne peut donc pas être réalisé.
Autrement dit, il n'est pas possible de relier 15 ordinateurs de sorte que chaque appareil soit relié avec exactement trois autres, car dans ce cas, la somme des degrés serait égale à $15 \times 3 = 45$ qui n'est pas pair.

3.1.9 On suppose que 6 invités sont là, et que Jack en fait partie. Parmi les 5 autres invités, soit il y en a 3 qui le connaissent, soit 3 ne le connaissent pas (par le principe des tiroirs). Sans perte de généralité, on peut supposer que ces trois invités connaissent Jack,

et qu'ils s'appellent respectivement Karen, Natalie et Billy. Deux possibilités alors : soit ces trois invités ne se connaissent pas (et on a notre trio gagnant), soit il y en a au moins deux qui se connaissent, et ils forment avec David le trio gagnant.



3.1.10

a) Le graphe est d'ordre 6 (6 sommets).

b) Le degré du sommet 1 : 3 ; du sommet 4 : 4.

c) Les sommets adjacents au sommet 2 : 1, 3, 4 et 6 ; au sommet 6 : 1, 2 et 5.

d) Les sommets adjacents à quatre sommets : 2 et 4.

3.1.11

longueur 3 : $stzy$;

longueur 4 : $stzxy$ et $svtzy$:

longueur 5 : $stzwxy$, $svtzxy$ et $svutzy$;

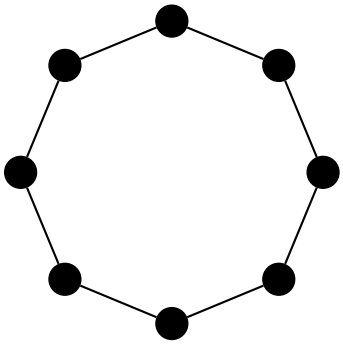
longueur 6 : $svutzxy$ et $svtzwxy$;

longueur 7 : $svutzwxy$.

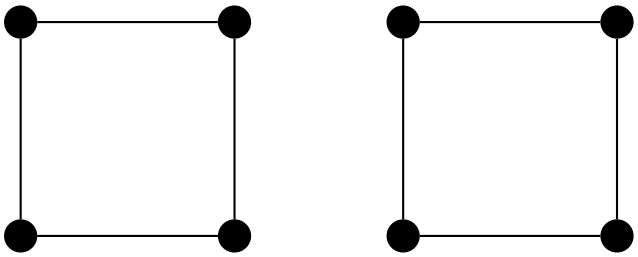
3.1.12

Il y a plusieurs possibilités, par exemple :

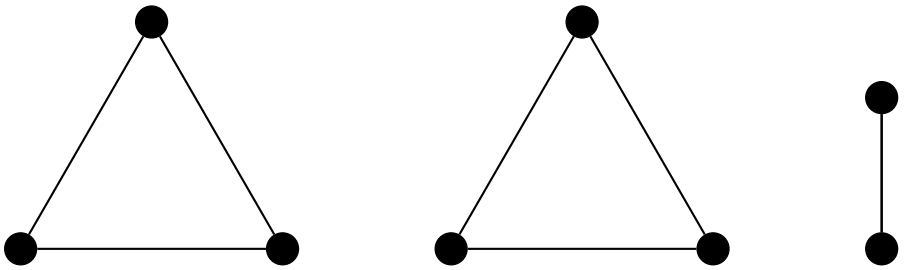
a)



b)



c)



3.1.13

a) Il n'est pas complet, car le sommet 1 n'est pas relié au sommet 5, par exemple.

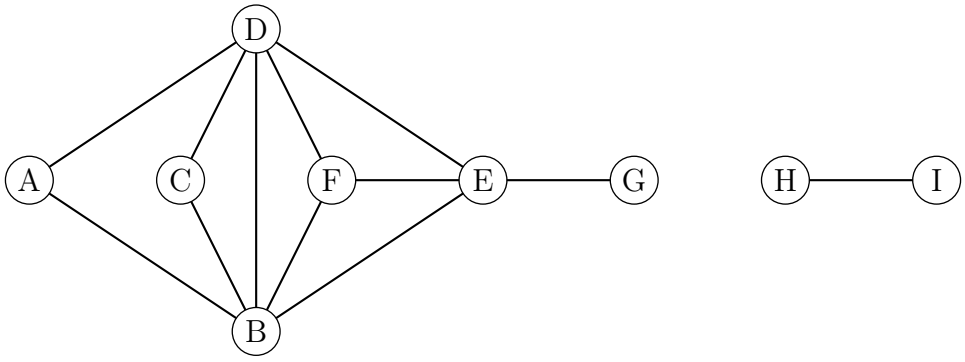
b) Il est connexe, car il est fait « d'un seul morceau ».

c) $G_1 = (2, 5, 3)$ et $G_2 = (2, 4, 3)$. Toute arête du graphe est un graphe complet d'ordre 2.

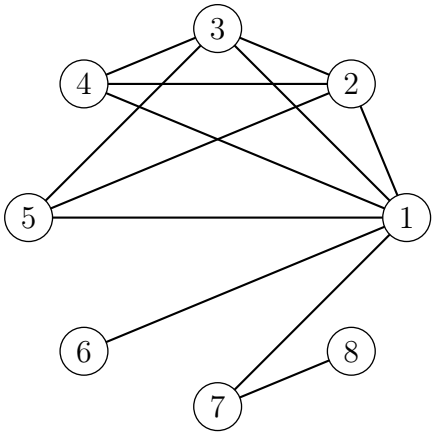
d) Le chemin est donné par la suite de sommets $(1, 2, 4, 3, 5)$; sa longueur est bien 4.

e) Il n'y a pas de cycle comprenant le sommet 1 vu que celui-ci est de degré 1.

3.1.14 Par exemple :



3.1.15 Par exemple :



3.1.16 C'est impossible, car tout graphe a un nombre pair de sommets de degré impair.

3.1.17 —

3.1.18 —

3.1.19 —

3.1.20 —

3.1.21 —

3.1.22 —

3.1.23 —

3.2.1 Les deux premiers ne sont pas des sous-graphes du graphe de Petersen, alors que le troisième est un sous-graphe.

3.2.2

a) Les sommets 1 et 4 sont de degré impair.

b) Il manque l'arête $6 - 4$.

c) $4 - 2 - 1 - 5 - 2 - 3 - 4 - 6 - 1$.

d) $4 - 2 - 1 - 5 - 2 - 3 - 4 - 6 - 1 - 4$.

3.2.3

a) On remarque que le graphe est connexe et que tous ses sommets sont de degré pair sauf les sommets D et G.

   Donc il existe une chaîne eulérienne entre les sommets D et G.

b) Comme on l'a vu dans la question précédente tous les sommets du graphe ne sont pas pairs donc il n'y a pas de cycle eulérien.

3.2.4 Le graphe de gauche n'est évidemment pas eulérien puisque non connexe. Celui du milieu est eulérien car tous les sommets sont de degré pair. Celui de droite est semi-eulérien, car seuls deux sommets sont de degré impair.

3.2.5 Le premier graphe admet une chaîne eulérienne, par exemple $D-B-C-D-E-A-B-E$. Le deuxième graphe admet un cycle eulérien: $F-G-H-I-J-F-H-J-G-I-F$. Le troisième graphe n'admet ni chaîne, ni cycle eulérien.

3.2.6 —

3.2.7 –

3.2.8 –

3.3.1 —

3.3.2 —

3.3.3 —

3.3.4 —

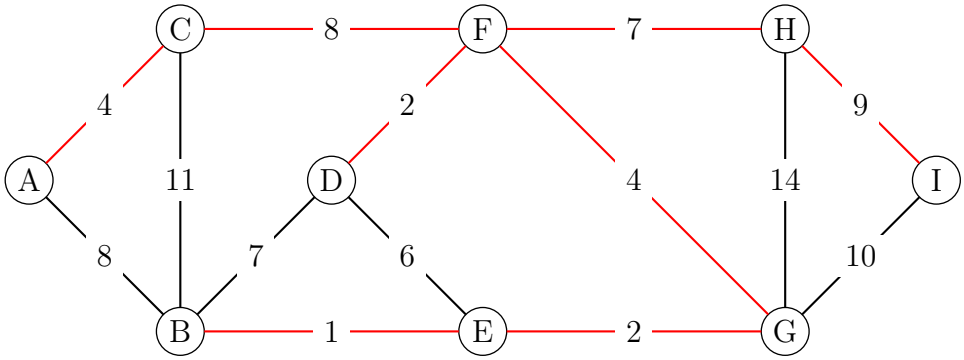
3.3.5 —

3.3.6 —

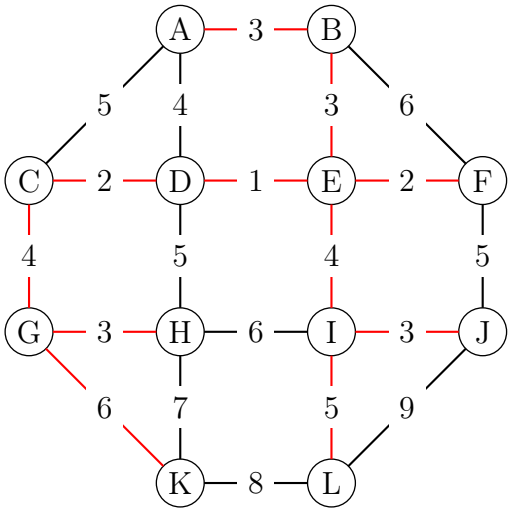
3.3.7 —

3.3.8 —

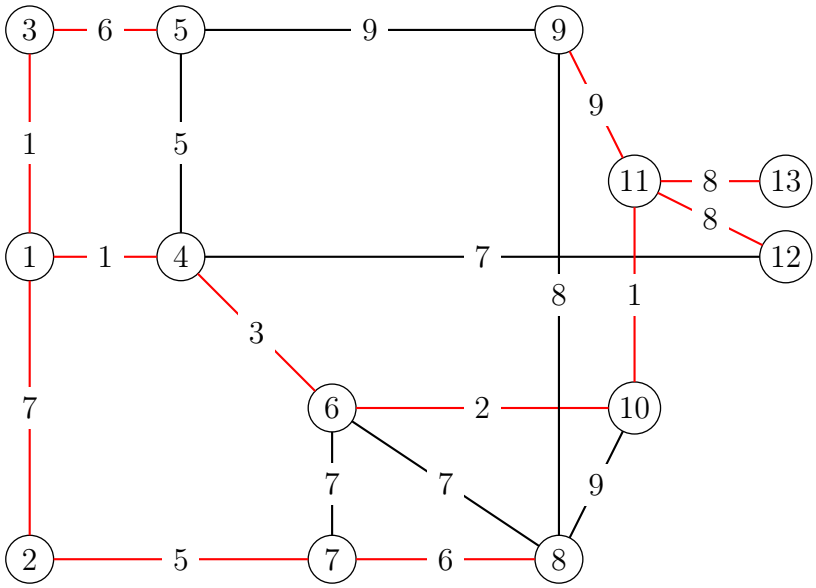
3.3.9



3.3.10



3.3.11 Par exemple, pour l'algorithme de Prim.



3.4.1

a) 4

b) $F - D - B$ de poids 8

3.4.2

Le chemin de poids minimal reliant A à E est $A - F - D - E$ de poids égal à 16.

3.4.3 $A - B - D - G$ de longueur 6

3.4.4 On obtient 3 chemins de 8 minutes :

a) $A - B - E - F$

b) $A - B - D - E - F$

c) $A - C - D - E - F$

3.4.5 Le chemin le plus court est $S - B - C - T$, de longueur 10.

3.4.6

S	T	B	A	C	D	E
0	∞	∞	∞	∞	∞	∞
—	∞	13_{S}	$\mathbf{7_S}$	28_{S}	∞	∞
—	∞	$\mathbf{11_A}$	—	28_{S}	32_{A}	17_{A}
—	∞	—	—	$\mathbf{16_B}$	17_{B}	17_{A}
—	∞	—	—	—	$\mathbf{17_B}$	17_{A}
—	22_{D}	—	—	—	—	$\mathbf{17_A}$
—	$\mathbf{22_D}$	—	—	—	—	—

Le chemin le plus court est S-A-B-D-T, de longueur 22.

3.4.7

S	A	B	C	D	E	F	T
0	∞	∞	∞	∞	∞	∞	∞
—	$\mathbf{1_S}$	3_{S}	6_{S}	∞	∞	∞	∞
—	—	$\mathbf{3_S}$	6_{S}	3_{A}	6_{A}	∞	∞
—	—	—	5_{B}	$\mathbf{3_A}$	6_{A}	6_{B}	∞
—	—	—	5_{B}	—	$\mathbf{4_D}$	6_{B}	10_{D}
—	—	—	$\mathbf{5_B}$	—	—	6_{B}	9_{E}
—	—	—	—	—	—	$\mathbf{6_B}$	9_{E}
—	—	—	—	—	—	—	$\mathbf{8_F}$

Le chemin le plus court est S-B-F-T, de longueur 8.

3.4.8

S	A	B	C	D	E	F	T
0	∞	∞	∞	∞	∞	∞	∞
—	$\mathbf{4_S}$	6_S	7_S	∞	∞	∞	∞
—	—	$\mathbf{6_S}$	7_S	11_A	9_A	∞	∞
—	—	—	$\mathbf{7_S}$	11_A	9_A	13_B	∞
—	—	—	—	11_A	$\mathbf{9_A}$	12_C	∞
—	—	—	—	$\mathbf{11_A}$	—	11_E	18_E
—	—	—	—	—	—	$\mathbf{11_E}$	18_E
—	—	—	—	—	—	—	$\mathbf{17_F}$

Le chemin le plus court est S-A-E-F-T, de longueur 17.

3.4.9

S	A	B	C	D	E	F	G	H	T
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
—	3_S	8_S	$\mathbf{2_S}$	∞	∞	∞	∞	∞	∞
—	$\mathbf{3_S}$	8_S	—	∞	3_C	4_C	∞	∞	∞
—	—	8_S	—	∞	$\mathbf{3_C}$	4_C	∞	∞	∞
—	—	8_S	—	∞	—	$\mathbf{4_C}$	9_E	∞	∞
—	—	8_S	—	$\mathbf{5_F}$	—	—	9_E	10_F	10_F
—	—	$\mathbf{8_S}$	—	—	—	—	9_E	9_D	10_F
—	—	—	—	—	—	—	$\mathbf{9_E}$	9_D	10_F
—	—	—	—	—	—	—	—	$\mathbf{9_D}$	10_F
—	—	—	—	—	—	—	—	—	$\mathbf{10_F}$

Le chemin le plus court est S-C-F-T, de longueur 10.

3.4.10

A	B	C	D	E	F	H	K
0	∞	∞	∞	∞	∞	∞	∞
—	$\mathbf{90_A}$	290_A	175_A	150_A	∞	∞	∞
—	—	275_B	175_A	$\mathbf{150_A}$	∞	∞	∞
—	—	275_B	$\mathbf{175_A}$	—	285_E	∞	∞
—	—	$\mathbf{275_B}$	—	—	280_D	395_D	∞
—	—	—	—	—	$\mathbf{280_D}$	395_D	535_C
—	—	—	—	—	—	$\mathbf{395_D}$	510_F
—	—	—	—	—	—	—	$\mathbf{510_F}$

Le chemin le plus court est A-D-F-K, de longueur 510.

3.4.11 Le plus court chemin est B-C-D-F-G. La longueur de ce chemin est de 36 minutes.

3.4.12 a) P–B–C–E–A–D–F–G

b) D'après le théorème d'Euler, il existe une chaîne eulérienne dans un graphe connexe si et seulement si exactement zéro ou deux de ses sommets sont de degrés impairs. Tous les sommets de ce graphe sont de degrés impairs sauf F, donc il n'existe aucun itinéraire qui emprunte une et une seule fois toutes les voies.

c) Le chemin le plus rapide de P vers G est d'une durée de 21 minutes : P-C-B-D-G.

3.4.13

Le chemin le plus court est X-N-A-C-F-D-H, de longueur 14.

Chapitre 4

Méthodes numériques

4.1 La bibliothèque matplotlib de Python : les fonctions de base

4.1.1 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np

plt.plot([1,3,4],[2,1,6])

plt.show()
```

4.1.2 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-2, 2, 100)
y = 2*x**2+3*x-4

plt.plot(x,y)
plt.show()
```

4.1.3 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np
from math import *

abscisses = np.linspace(-4,4,100)
ordonnées = [cos(x)+3*sin(2*x) for x in abscisses]

plt.plot(abscisses,ordonnées)
plt.show()
```

4.1.4 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-4,4,100)
y = np.cos(x)+3*np.sin(2*x)

plt.plot(x,y)
plt.show()
```

4.1.5 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-1,3,100)
y = -2*x+3
plt.plot(x,y)
y = x**2 - 4*x + 4

plt.plot(x,y)
plt.show()
```

4.1.6 Copier ces lignes dans une fenêtre de Python et observer le résultat.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-np.pi,np.pi,100)
for n in range(1,5):
    y = np.cos(n*x)
    plt.plot(x,y, label="n="+str(n))

plt.legend(loc="lower right")
plt.show()
```

4.1.7 Afficher la représentation graphique de la fonction $f(x) = x^3 + 3x^2 - 9x + 1$ pour x entre -2 et 4.

4.1.8 Afficher la représentation graphique de la fonction $f(x) = 3\cos(2\,x) - 2\sin(3\,x)$ pour x entre $-\pi$ et π.

4.1.9 Afficher les représentations graphiques de la famille de fonctions $f(x) = \dfrac{1 - n\,x}{x - 1}$ pour n allant de -3 à 3. On se placera dans un repère allant de -2 à 4 pour les abscisses et de -8 à 8 pour les ordonnées. On fera apparaitre aussi une légende en haut à gauche.

4.1.10

a) Tracer en vert la fonction qui à x associe $log(x)$ pour x variant de 0.1 à 10 avec un pas de 0.1. Utiliser la fonction `np.log()`.
Ajouter les légendes et le titre appropriés.

b) Tracer en rouge la fonction qui à x associe $\sqrt{x}$ pour x variant de 0.1 à 10 avec un pas de 0.1. Utiliser la fonction `np.sqrt()`.
Ajouter les légendes et le titre appropriés.

c) Selon vous quelle est la fonction qui croit le plus vite en $+\infty$? qui décroit le plus vite en 0 ?
Vérifier en superposant les 2 courbes.

d) Tracer `plt.plot(x,np.exp(np.log(x)))`. Quelle est la forme de la courbe obtenue ? Pourquoi obtient-on cette forme ?

4.1.11 Tracer la spirale d'Archimède donné par le système d'équations paramétriques.

$$\begin{cases} x(t) &= t\,\cos(t) \\ y(t) &= t\,\sin(t) \end{cases}$$

4.1.12 Tracer la courbe de la fonction f dans l'intervalle I ainsi que sa tangente au point d'abscisse a.

a) $f(x) = x^2$, $I = [-1\,;3]$, $a = 1$.

b) $f(x) = e^x - 3\,x^2$, $I = [-1\,;5]$, $a = 4$.

c) $f(x) = \sin(x) - \dfrac{x}{4}$, $I = [-1\,;\pi]$, $a = 2$.

4.2 Zéros de fonctions : méthode de la bissection

4.2.1 Soit la fonction $f(x) = (x+2)(x+1)(x-1)$.

a) Représenter la fonction dans l'intervalle $I = [0.1\,;2.4]$.

b) Écrire les six premiers intervalles successifs obtenus par la méthode de la bissection pour chercher les zéros de la fonction $f(x)$ dans l'intervalle I.

4.2.2 Pour chacune des fonctions suivantes, faire trois itérations de la méthode de la bissection à partir de l'intervalle indiqué pour trouver sa racine dans cet intervalle. Puis déterminer le nombre d'itérations nécessaires pour obtenir une solution dont le chiffre des millièmes est significatif.

a) $f(x) = 0.3\,x^2 - 0.6\,x - 0.8$ dans l'intervalle $[2.8\,;3]$.

b) $f(x) = x^5 + x^3 - 1$ dans l'intervalle $[0\,;1]$.

c) $f(x) = (x\,\cos(x))^2 - 2.3$ dans l'intervalle $[1.3\,;5.3]$.

4.2.3 On considère l'équation :

$$e^x - (x+5) = 0$$

a) Déterminer le nombre et la position approximative des solutions positives de cette équation.

b) Utiliser l'algorithme de la bissection pour déterminer chacune de ces racines avec une erreur absolue inférieure à 10^{-7}.

4.2.4 Montrer que chacune des fonctions f suivantes possède un zéro unique. Calculer ce zéro par l'algorithme de la bissection.

a) $f(x) = x^3 + 10$
b) $f(x) = x^5 - 2\,x^4 + 100\,x^3 - 2$
c) $f(x) = x - 1 + \frac{1}{2}\,\sin(x)$
d) $f(x) = 3\,x - \cos(x)$
e) $f(x) = \ln x - \cos(x)$
f) $f(x) = x + e^x$

4.2.5 Montrer que la fonction $f = x^3 + 2\,x - 1$ possède un zéro dans l'intervalle $I = [0\,;1]$. Calculer ce zéro par l'algorithme de la bissection.

4.2.6 Déterminer une valeur approchée à 10^{-6} près des éventuels zéros de la fonction

$$f(x) = x^7 + 23\,x^5 + 2\,x^4 - 2$$

Calculer ce zéro par l'algorithme de la bissection.

4.2.7 Utiliser la méthode de la dichotomie pour obtenir une valeur approchée du nombre $\sqrt[3]{2}$ à 10^{-4} près.

4.2.8 Utiliser la méthode de la dichotomie pour obtenir une valeur approchée du nombre π.

4.3 Zéros de fonctions : méthode de Newton

4.3.1 Soit la fonction

$$f(x) = -5\,x^3 + 7\,x^2 + 3\,x - 3$$

a) Calculer les coordonnées des points à tangente horizontale et du point d'inflexion. Tracer le graphe de f dans l'intervalle $[-1\,;2]$.
b) Calculer la valeur des premiers termes de la suite de Newton si l'on choisit l'estimation initial $x_0 = 1$.
c) Calculer la valeur des premiers termes de la suite de Newton si l'on choisit l'estimation initial $x_0 = 2$.
d) Déterminer l'unique zéro de f compris entre 1 et 2.

4.3.2 L'équation

$$e^x - 3\,x^2 = 0$$

possède les solutions $s_1 = -0.4589623$ et $s_2 = 0.91$ ainsi qu'une troisième solution s_3 située près de 4.
Utiliser la méthode de Newton pour déterminer s_3 avec six décimales.

4.3.3 Trouver une estimation d'un zéro strictement positif des fonctions suivantes en appliquant la méthode de Newton.

a) $f(x) = x^3 - 2$ b) $f(x) = \sin(x) - \dfrac{x}{4}$

4.3.4 Soit la fonction

$$f(x) = x^3 - 2\,x - 5$$

a) Montrer que l'équation $f(x) = 0$ admet une unique solution α sur $\mathbb{R}$. Montrer que $2 < \alpha < 3$.

b) A partir de $x_0 = 1$, déterminer le zéro de la fonction.

4.3.5 Donner une approximation de $\sqrt{2}$ avec la méthode de Newton en partant du point $x_0 = 2$.

4.3.6 Une des difficultés de la méthode de Newton est de bien choisir x_0 de sorte que l'on ne tombe pas sur un point où la dérivée s'annule.
Utiliser l'algorithme de Newton avec la fonction $f(x) = \cos(x)$ en partant des points d'abscisse 1, 0.5, 0.1 puis 0.01.
Commenter et expliquer la situation à l'aide d'un graphique.

4.4 Zéros de fonctions : méthode de la sécante

4.4.1 A l'aide de la méthode de la sécante, calculer la racine douzième de 1.1 avec une précision de 8 décimales.

4.4.2 Rechercher, en utilisant la méthode de la sécante, les solutions de l'équation

$$x^3 + 1 = 3\,x$$

4.4.3 Calculer dans l'intervalle $I = [0\,;\pi]$ une approximation de la solution de l'équation

$$\cos(x) = 2\,\sin(x)$$

4.4.4 Rechercher, en utilisant la méthode de la sécante, les solutions de l'équation

$$e^{-x} = -\ln(x)$$

4.5 Un peu d'intégration numérique

4.5.1 Sans utiliser de bibliothèque autre que la bibliothèque mathématique standard de python, écrire une fonction qui prend deux nombres réels $a < b$ et un nombre entier n en paramètres. Cette fonction doit renvoyer une liste de $n + 1$ valeurs réelles, notées $x_0, x_1, \ldots, x_n$ telles que $x_0 = a$, $x_n = b$, tous les autres nombres étant uniformément répartis dans l'intervalle $[a, b]$.
L'instruction

```
print(listePoints(0, 2, 10))
```

fait afficher la liste

```
[0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2]
```

à quelques erreurs d'arrondi près.

4.5.2 Compléter le code de la fonction de l'exercice précédent en lui ajoutant un paramètre optionnel `aleatoire=False`. Par défaut, le comportement de la fonction est le même que celle de l'exercice précédent. Si le paramètre `aleatoire` prend la valeur `True`, la fonction doit renvoyer une liste de $n+1$ valeurs réelles, notées $x_0, x_1, \ldots, x_n$ telles que $x_0 = a$, $x_n = b$, tous les autres nombres étant aléatoirement choisis dans l'intervalle $[a, b]$.

L'instruction

```
print(listePoints(0, 2, 10)) # Valeur par défaut de aleatoire
```

fait afficher la liste

```
[0, 0.2, 0.4,
 0.6000000000000001, 0.8, 1.0, 1.2, 1.4,
 1.5999999999999999, 1.7999999999999998,
 1.9999999999999998]
```

L'instruction

```
print(listePoints(0, 2, 10, True))
```

fait afficher la liste

```
[0,
 0.06479694651588885, 0.28981342176727964, 0.3179007067671271,
 0.4070274425545328,  1.015739468424949,   1.1182123677493638,
 1.2618287932253878,  1.473438685747842,   1.519412756197574,
 2]
```

Il faut noter que le résultat change à chaque fois que le programme tourne.

4.5.3 Créer une classe `Fonction` qui stocke dans un champ une fonction mathématique f. Dans cette classe, créer une méthode dont le nom est `listeValeurs` qui prend une liste de nombres en paramètre et qui renvoie la liste des valeurs prises par la fonction f en chacun des points.

Importer le module `math` de python et vérifier que l'exécution des lignes

```
lp = listePoints(0, math.pi / 2, 10)
    f = Fonction(math.sin)
    for val in f.listeValeurs(lp):
        print(val)
```

donne l'affichage suivant :

```
0.0
0.15643446504023087
0.3090169943749474
0.45399049973954675
0.5877852522924731
0.7071067811865475
0.8090169943749473
0.8910065241883678
0.9510565162951535
0.9876883405951378
1.0
```

4.5.4 Compléter la classe `Fonction` avec une méthode dont l'en-tête est

```
def sommeRect(self, a, b, n, aleatoire=False)
```

qui renvoie

- la liste des n points choisis dans l'intervalle $[a, b]$;
- une liste de valeurs définies par l'expression

$$\min(f(x_i); f(x_{i+1}))$$

  pour i compris entre 0 et $n-1$.
- Une approximation de

$$\int_a^b f(x)\,dx$$

  par une somme d'aires de rectangles, en sélectionnant à chaque itération le rectangle « sous la courbe ».
- une liste de valeurs définies par l'expression

$$\max(f(x_i); f(x_{i+1}))$$

  pour i compris entre 0 et $n-1$.
- Une approximation de

$$\int_a^b f(x)\,dx$$

  par une somme d'aires de rectangles, en sélectionnant à chaque itération le rectangle « en dessus de la courbe ».

La valeur par défaut du paramètre `aleatoire` donne un découpage régulier de l'intervalle $[a, b]$. Si la valeur `True` est passée en paramètre, le découpage est aléatoire.

Faire calculer

$$\int_0^2 (2\,x^2 - x^3)\,dx$$

en découpant l'intervalle en 10 morceaux, uniformément ou aléatoirement.
Écrire les instructions qui donnent l'affichage suivant lorsque le programme tourne :

```
Intervalle découpé en 10 morceaux identiques :

1.0848000000000002
1.5552

Intervalle découpé aléatoirement en 10 morceaux (5 essais)

0.8943893487419508
1.7009476901969376
*
0.8878935584553052
1.770292543499321
*
0.8352384010982953
1.5033073760799847
*
0.9881561937421527
1.5762026218159841
*
0.8617505588265767
1.7164006543400632
```

4.5.5 À l'aide de la méthode `sommeRect()` de la classe `Fonction`, calculer une approximation de

$$\int_{-0.5}^{1} e^{-x^2}\,dx$$

dont les 6 premières décimales sont correctes, en découpant l'intervalle $[-0.5, 1]$ en n morceaux. Quelle est la plus petite valeur de n qui garantit ce résultat ?

4.5.6 Pour $n \in \{10, 100, 1000, 10000, 100000\}$, faire calculer une approximation de

$$\int_{1}^{2} \frac{1}{x}\,dx$$

uniforme et aléatoire.

a) Comparer les valeurs obtenues par les deux méthodes.

b) Comparer avec la valeur exacte obtenue par le calcul de la primitive.

Quelle conclusion en tirer ?

4.5.7 Compléter la classe `Fonction` avec une méthode dont l'en-tête est

```
def sommeTrap(self, a, b, n, aleatoire=False)
```

qui renvoie une approximation de

$$\int_{a}^{b} f(x)\,dx$$

à l'aide d'une somme d'aires de trapèzes. La somme peut être calculée de la façon suivante:

Pour tout i compris entre 0 et $n-1$, l'expression

$$(x_{i+1} - x_i) \cdot \frac{(f(x_i) + f(x_{i+1}))}{2}$$

contribue à la somme totale.

4.5.8 À l'aide de la méthode `sommeTrap()` de la classe `Fonction`, calculer une approximation de

$$\int_{-0.5}^{1} e^{-x^2}\, dx$$

dont les 6 premières décimales sont correctes, en découpant l'intervalle $[-0.5, 1]$ en n morceaux. Quelle est la plus petite valeur de n qui garantit ce résultat ?

4.5.9 Compléter la classe `Fonction` avec une méthode dont l'en-tête est

```
def sommeSim(self, a, b, n, aleatoire=False)
```

qui renvoie une approximation de

$$\int_a^b f(x)\, dx$$

à l'aide d'une somme d'aires, calculées de la façon suivante :

- L'intervalle $[a, b]$ est découpé en n sous-intervalles, régulièrement ou non.
- On détermine pour chaque sous-intervalle $I_i = [x_i, x_{i+1}]$ l'expression mathématique de la parabole qui passe par les trois points $(x_i, f(x_i))$, $(m, f(m))$ et $(x_{i+1}, f(x_{i+1}))$ où m désigne le milieu de l'intervalle I_i.
- On calcule ensuite la valeur exacte de l'intégrale de cette parabole sur l'intervalle I_i, qui contribue à la somme totale.
- Ce calcul ayant été fait pour tout i, on renvoie la somme qui donne l'approximation de l'intégrale.

4.6 Solutions des exercices

4.1.1 —

4.1.2 La fonction `np.linspace(debut, fin, nombre)` permet de créer une liste de `N` nombres qui commencent à la valeur `debut` et s'arrête à la valeur `fin` et uniformément répartis.

4.1.3 Tracé de $y = \cos(x) + 3\,\sin(2\,x)$

4.1.4 Tracé de $y = \cos(x) + 3\,\sin(2\,x)$

4.1.5 —

4.1.6 —

4.1.7

```
import matplotlib.pyplot as plt
import numpy as np
def f(x):
    return x**3+3*(x**2)-9*x+1
x = np.linspace(-2,4,100)
plt.plot(x,f(x))
plt.grid(True)
plt.title("Exercice 5.1.7")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```

4.1.8

```
import matplotlib.pyplot as plt
import numpy as np
import math
pi = math.pi
```

```
def f(x):
    return 3*np.cos(2*x)-2*np.sin(3*x)
x = np.linspace(-pi,pi,100)
plt.plot(x,f(x))
plt.grid(True)
plt.title("Exercice 5.1.8")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```

4.1.9

Il faut faire attention à ne pas représenter la droite $x = 1$.

```
import matplotlib.pyplot as plt
import numpy as np
def f(x, n):
    return (1-n*x)/(x-1)
x = np.linspace(-2,4,201)
for n in range(-3, 4):
    l = 'n = ' + str(n)
    plt.plot(x,f(x, n), label=l)
plt.grid(True)
plt.title("Exercice 5.1.9")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.xlim(-2, 4)
plt.ylim(-8, 8)
plt.legend()
plt.show()
```

4.1.10

```
import matplotlib.pyplot as plt
```

```
import numpy as np
x = np.linspace(0.1,10,100)
plt.plot(x,np.log(x),color="green",label="log(x)")
plt.plot(x,np.sqrt(x),color="red",label="sqrt(x)")
plt.plot(x,np.exp(np.log(x)),color="blue")
plt.grid(True)
plt.title("Exercice 5.1.10")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()
plt.show()
```

On a $e^{\ln(x)} = x$.

4.1.11

```
import matplotlib.pyplot as plt
import numpy as np
t = np.linspace(0, 10 * np.pi, 300) # Spirale d'Archimede
x = t * np.cos(t)
y = t * np.sin(t)
plt.plot(x, y)
plt.show()
```

4.1.12

Par exemple :

```
import matplotlib.pyplot as plt
import numpy as np
def f(x):
    return x*x
def df(x):
    return 2*x
x = np.linspace(-1,3,100)
```

```
y = df(1)*(x-1)+f(1)
plt.plot(x,f(x),color="green",label="x^2")
plt.plot(x,y,color="red",label="tangente")
plt.grid(True)
plt.title("Exercice 5.1.12")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()
plt.show()
```

4.2.1

$I_0 = [0.1\,;2.4]\,; I_1 = [0.1\,;1.25]\,; I_2 = [0.675\,;1.25]\,; I_3 = [0.9625\,;1.25]\,; I_4 = [0.9625\,;1.10625]\,;$
$I_5 = [0.9625\,;1.034375]$

4.2.2

a) 2.9 ; 2.95 ; 2.925. Il faut au moins neuf itérations.

b) 0.5 ; 0.75 ; 0.875. Il faut au moins onze itérations.

c) 3.3 ; 2.3 ; 1.8. Il faut au moins treize itérations.

4.2.3

a) Si $f(x) = e^x - (x+5)$, alors $f'(x) = e^x - 1$ est positive pour $x > 0$ et donc f est strictement croissante sur $\mathbb{R}^+$. Par conséquent, f ne croise l'axe des x qu'une seule fois. Puisque $f(0) = -4$ et $f(2) = 0.38$, il y a une seule racine entre $x = 0$ et $x = 2$.

b) On obtient $x = 1.9368473291$ en 23 itérations à partir de l'intervalle $[0\,;2]$.

4.2.4

a) -2.1544347

b) 0.2718682

c) 0.6840367

d) 0.3167508

e) 1.302964

f) -0.5671433

4.2.5

$x = 0.4533976515$

4.2.6

$x = 0.595471$

4.2.7 —

4.2.8 —

4.3.1

a) minimum : $(-0.18\,;-3.28)$, maximum : $(1.11\,;2.12)$, point d'inflexion : $(0.47\,;-0.58)$

b) 0, 1, 0, 1, 0, 1, 0, ...

c) 1.6896552, 1.5618930, 1.5372456, 1.5363390, 1.5363378

d) 1.5363378

4.3.2

$s_3 = 3.733079$

4.3.3

a) 1.259921050

b) 2.474576794

4.3.4

a) A partir du tableau des variations de f

b) 2.0945514815

4.3.5 —

4.3.6 —

4.4.1

$r = 1,00796779$

4.4.2

$x_1 = -1.8793852416$, $x_2 = 0.3472963553$ et $x_3 = 1.5320888862$

4.4.3

$x = 0.4636476092$

4.4.4

$x = 0.5671432873$

4.5.1

```
def listePoints(a, b, n):
    if a > b:
        a, b = b, a
    lp = [a]
    delta = (b - a) / n
    for i in range(n):
        a = a + delta
        lp.append(a)
    return lp
```

4.5.2

```
import random

def listePoints(a, b, n, aleatoire=False):
    if a > b:
        a, b = b, a
    lp = [a]
    if not aleatoire:
        delta = (b - a) / n
        for i in range(n):
            a = a + delta
            lp.append(a)
    else:
        for i in range(n - 1):
            x = a + (b - a) * random.random()
```

```
            lp.append(x)
        lp.append(b)
        lp.sort()
    return lp
```

4.5.3

```
import math
import random



def listePoints(a, b, n, aleatoire=False):
    if a > b:
        a, b = b, a
    lp = [a]
    if not aleatoire:
        delta = (b - a) / n
        for i in range(n):
            a = a + delta
            lp.append(a)
    else:
        for i in range(n - 1):
            x = a + (b - a) * random.random()
            lp.append(x)
        lp.append(b)
        lp.sort()
    return lp



class Fonction(object):
    def __init__(self, f):
        self.f = f
```

```
    def listeValeurs(self, lp):
        lv = []
        for x in lp:
            lv.append(self.f(x))
        return lv


if __name__ == "__main__":
    lp = listePoints(0, math.pi / 2, 10)
    f = Fonction(math.sin)
    for val in f.listeValeurs(lp):
        print(val)
```

4.5.4

```
import math
import random

def listePoints(a, b, n, aleatoire=False):
    if a > b:
        a, b = b, a
    lp = [a]
    if not aleatoire:
        delta = (b - a) / n
        for i in range(n):
            a = a + delta
            lp.append(a)
    else:
        for i in range(n - 1):
            x = a + (b - a) * random.random()
            lp.append(x)
```

```
        lp.append(b)
        lp.sort()
    return lp


class Fonction(object):
    def __init__(self, f):
        self.f = f

    def listeValeurs(self, lp):
        lv = []
        for x in lp:
            lv.append(self.f(x))
        return lv

    def sommeRect(self, a, b, n, aleatoire=False):
        lp = listePoints(a, b, n, aleatoire)
        sSup, sInf = 0.0, 0.0
        lvSup = []
        lvInf = []
        for i in range(n):
            delta = lp[i + 1] - lp[i]
            valeurs = self.f(lp[i]), self.f(lp[i + 1])

            v_inf = min(valeurs)
            lvInf.append(v_inf)
            sInf += v_inf * delta

            v_sup = max(valeurs)
            lvSup.append(v_sup)
            sSup += v_sup * delta
```

```
        return lp, lvSup, sSup, lvInf, sInf

if __name__ == "__main__":
    g = Fonction(lambda x : 2 * x**2 - x**3)
    lp, lvSup, sSup, lvInf, sInf = g.sommeRect(0.0, 2.0, 10)
    print("Intervalle découpé en 10 morceaux identiques : ")
    print()
    print(sInf)
    print(sSup)
    print()
    print("Intervalle découpé aléatoirement en 10 morceaux (5 essais)")
    print()
    for i in range(5):
        lp, lvSup, sSup, lvInf, sInf = g.sommeRect(0.0, 2.0, 10, True)
        # print(lv)
        print(sInf)
        print(sSup)
        print("*")
```

4.5.5

```
if __name__ == "__main__":
    h = Fonction(lambda x : math.e**(-x**2))
    n = 100
    lp, lvSup, sSup, lvInf, sInf = h.sommeRect(-0.5, 1.0, n)
    print("Intervalle découpé en {} morceaux identiques : ".format(n))
    print()
    print(sInf)
    print(sSup)
    print()
    while sSup - sInf > 0.000001:
```

```
        n *= 10
        lp, lvSup, sSup, lvInf, sInf = h.sommeRect(-0.5, 1.0, n)
    print(sSup)
    print(sInf)
    print(n)
    print("Valeur donnée par le site Wolfram Alpha")
    print("1.208105139225219474155170372872306108438258775776897 4")

if __name__ == "__main__":
    aire = 0.693147180559836
    g = Fonction(lambda x : 1 / x)
    nombres = [10, 100, 1000, 10000, 100000]
    for n in nombres:
        lp, lvSup, sSup, lvInf, sInf = g.sommeRect(1.0, 2.0, n)
        lpa, lvSupa, sSupa, lvInfa, sInfa = g.sommeRect(1.0, 2.0, n, True)
        print(sInf, sInfa, sSup, sSupa)
        print(5 * "*")
        lpa, lvSupa, sSupa, lvInfa, sInfa = g.sommeRect(1.0, 2.0, n, True)
    print(5 * "*")
    print(aire)

def sommeTrap(self, a, b, n, aleatoire=False):
    lp = listePoints(a, b, n, aleatoire)
    s = 0.0
    lv = []
    for i in range(n):
        delta = lp[i + 1] - lp[i]
        valeur = (self.f(lp[i]) + self.f(lp[i + 1])) * 0.5

        lv.append(valeur)
        s += delta * valeur
```

```
    return lp, lv, s
```

4.5.6 4.5.7 4.5.8

```
if __name__ == "__main__":
    h = Fonction(lambda x : math.e**(-x**2))
    n = 100
    lp_1, lv_1, s_1 = h.sommeTrap(-0.5, 1.0, n)
    n *= 10
    lp_2, lv_2, s_2 = h.sommeTrap(-0.5, 1.0, n)
    while abs(s_2 - s_1) > 0.000001:
     lp_1, lv_1, s_1 = h.sommeTrap(-0.5, 1.0, n)
        n *= 10
        lp_2, lv_2, s_2 = h.sommeTrap(-0.5, 1.0, n)
    print(s_2)
    print(n)
    print("Valeur donnée par le site Wolfram Alpha")
    print("1.208105139225219474155170372872306108438258775776897 4")

# Deux fonctions auxiliaires
def coeffParaboleParTroisPoints(x_1, y_1, x_2, y_2, x_3, y_3):
    a = ((y_3 - y_1)/(x_3 - x_1) - (y_2 - y_1)/(x_2 - x_1)) / (x_3 - x_2)
    b = -a * (x_2 + x_1) + (y_2 - y_1)/(x_2 - x_1)
    c = y_1 - a * x_1**2 - b * x_1
    return a, b, c


def primPara(a, b, c):
    return lambda x: 1/3 * a * x**3 + 1/2 * b * x**2 + c * x


# Méthode de la classe Fonction
def sommeSim(self, a, b, n, aleatoire=False):
    lp = listePoints(a, b, n, aleatoire)
    s = 0.0
```

```
    for i in range(n):
        x_3 = lp[i + 1]
        y_3 = self.f(x_3)
        x_2 = (lp[i + 1] + lp[i]) / 2
        y_2 = self.f(x_2)
        x_1 = lp[i]
        y_1 = self.f(x_1)
        a, b, c = coeffParaboleParTroisPoints(x_1, y_1, x_2, y_2, x_3, y_3)
        F = primPara(a, b, c)
        s += F(x_3) - F(x_1)

    return lp, s
```