RÉPUBLIQUE FRANÇAISE
*Liberté*
*Égalité*
*Fraternité*

UNIVERSITÉ CAEN NORMANDIE

CONTRÔLE CONTINU GRAPHES

# Graphes, recherche arborescente et complexité

Noms et Prénoms de l'étudiant :

OROU-GUIDOU Amirath Fara

Numéro de l'étudiants :

22012235

Parcours : M1 IA Science de Données et Santé

Groupe : 1

Sous la supervison :

TERRIER Véronique

Septembre 2023

# Table des matières

# 1 Un problème dans la classe P

On considère le problème de décision **ZERO** qui consiste à déterminer si une liste d'entiers contient ou non trois éléments dont la somme est nulle.

| **Problème ZERO** |
| --- |
| **Donnée** Une liste d'entiers E<br>**Question** Existe-t-il a, b, c trois éléments de E distincts tels que a + b + c = 0 ? |

## 1.1 Question 1.

**1.a.** La liste [6, -7, -11, 12, 10, -18, 5, 3] est-elle une instance positive de ZERO ? Justifiez votre réponse.

**Réponse :**

Oui la liste est une instance positive de ZERO car il existe trois éléments a, b, c de la liste avec a = 6, b = 12, c = -18 tel que 6 + 12 - 18 = 0.

**1.b.** La liste [-10, 13, -7, -2, -4] est-elle une instance positive de ZERO ? Justifiez votre réponse.

**Réponse :**

Non la liste n'est pas une instance positive de ZERO. C'est une instance négative car lorsqu'on parcourt tous les éléments de la liste et on fait la somme de trois éléments distinctes de la liste, on obtient pas zéro.

Pour résoudre le problème de décision ZERO, la première idée est d'essayer tous les tuples (a, b, c) possibles d'entiers distincts contenus dans E.

## 1.2 Question 2.

**2.a** Ecrivez une fonction **naïfZERO** qui résout le problème ZERO en mettant en œuvre cette approche naïve.

**Réponse :**

```
def naifZERO(E):
    //n : taille de la liste
    n=len(E)
    for i in range(n): // Complexité en O(n)
        for j in range(i+1, n): // Complexité en O(n)
            for k in range(j+1, n): // Complexité en O(n)
                if E[i]+E[j]+E[k]==0:
                    return True
    return False
```

**2.b** Donnez la complexité de votre procédure **naifZERO** en fonction du nombre d'éléments de E ? Justifiez votre réponse.

**Réponse :**

La complexité de cet algorithme est : $\mathbf{O}(n^3)$.

**Justification** : Nous avons **3** boucles imbriquées. Chaque boucle est exécutée en fonction de la taille de l'entrée qui est **n**, il y a **n itétérations** dans chaque boucle. La première boucle parcourt les indices de **i** jusqu'à **n**, la deuxième parcourt les indices de **j** allant de **i+1** à **n** et la troisième parcourt les indices de **k** allant de **j+1** à **n**. Le nombre total d'opérations en multipliant le nombre d'itérations de chaque boucle est **n*n*n** $= n^3$. D'où la complexité est $\mathbf{O}(n^3)$. Cette complexité est **Polynomiale**

## 1.3 Question 3.

**3.a** Écrivez une fonction testZERO qui résout le problème ZERO.

**Réponse :**

```
def testZERO(E):
    E.sort() // On trie la liste. Complexité en 0(nlogn)
    n=len(E) // La taille de la liste
    for i in range(0, n-3): // Complexité en O(n)
        deb = i+1
        fin = n-1
        while deb < fin: // Complexité en O(n)
            if E[i]+E[deb]+E[fin] > 0:
                fin -= 1
            elif E[i]+E[deb]+E[fin] < 0:
                deb += 1
            else:
                return True
    return False

```

**3.b** Donnez la complexité de votre procédure testZERO en fonction du nombre d'éléments de E ? Justifiez soigneusement votre réponse.

**Réponse :**

La complexité de cet algorithme est : $\mathbf{O}(n^2)$

**Justification** : Pour le trie de la liste, on a une complexité en **O(nlogn)** ; on parcourt chaque élément de la liste en **O(n)** et dans le pire des cas, on va parcourir du suivant de cet élément de la liste jusqu'à la fin de la liste ce qui nous donne une complexité en $\mathbf{O}(n^2)$. En conclusion, on a une complexité en **O(nlogn)** + $\mathbf{O}(n^2)$ avec **n** la taille de la liste **E**. Or **O(nlogn)** < $\mathbf{O}(n^2)$, donc **O(nlogn)** + $\mathbf{O}(n^2) \simeq O(n^2)$. D'où la complexité est en $\mathbf{O}(n^2)$.

# 2 Réduction de TRILISTE à ZERO

On envisage maintenant une variante du problème **ZERO**.

| **Problème TRILISTE** |
|---|
| **Donnée** Trois listes d'entiers X, Y et Z<br>**Question** Existe-t-il $x \in X$, $y \in Y$ et $z \in Z$ tels que x + y + z = 0 ? |

## 2.1 Question 4.

Donnez, en justifiant vos réponses ,

**4.a.** un exemple d'instance positive $(X_p, Y_p, Z_p)$ du problème TRILISTE, de taille au moins 8 ($|X_p| + |Y_p| + |Z_p| \geq 8$)

**Réponse :**

On prend :

$$\begin{cases} X_p = [2, -4, -7] \\ Y_p = [-5, 8, -6, -1] \\ Z_p = [-3, -10, 9] \end{cases}$$

$\forall x \in Xp, y \in Yp$ et $z \in Zp$ on a pour :
x = 2 , y = 8 et z = -10 ; x + y + z = 0 d'où $(X_p, X_p, Z_p)$ est une instance positive du problème TRILISTE.

**4.b.** un exemple d'instance négative $(X_n, Y_n, Z_n)$ de taille au moins 6

**Réponse :**

On prend :

$$\begin{cases} X_n = [2, 4, 7] \\ Y_n = [5, 8, 6] \\ Z_n = [3, 10, 9] \end{cases}$$

$\forall x \in X_n, y \in Y_n$ et $z \in Z_n$, il n'existe pas de x, y, z tels que x + y + z = 0. De plus, tous les éléments de chaque liste sont positives donc on ne peut pas avoir une instance positive de ZERO à moins qu'on a un **0** dans chaque liste.

On considère la transformation, notée **r**, des instances TRILISTE en instances ZERO :

D'une instance **TRILISTE** composée de trois listes d'entiers

$$(X, Y, Z)$$

on définit l'instance **ZERO**

r(X, Y, Z) $= [10x + 1 : x \in X] + [10y + 2 : y \in Y\ ] + [10z - 3 : z \in Z]$

## 2.2 Question 5.

**Question :** Ecrivez la fonction reduction qui, à partir d'une instance TRILISTE (trois listes d'entiers) construit l'instance ZERO correspondante (une liste d'entiers).

**Réponse :**

```
def reduction(X, Y, Z):
    // Définition d'une liste vide
    liste = []
    for x in X:  // Complexité en O(|X|)
        liste.append(10*x + 1)             // O(1)
    for y in Y:  // Complexité en O(|Y|)
        liste.append(10*y + 2)             // O(1)
    for z in Z:  // Complexité en O(|Z|)
        liste.append(10*z - 3)             // O(1)
    return liste

```

## 2.3 Question 6.

**6.a.** Donnez la complexité du calcul de reduction en fonction de n $= |X| + |Y\ | + |Z|$ , taille de l'entrée (X, Y, Z) .

**Réponse :**

Soit **n** la somme de la taille de chaque liste ; $\mathbf{n} = |\mathbf{X}| + |\mathbf{Y}| + |\mathbf{Z}|$.
On parcourt les trois listes X, Y et Z une seule fois, et on fait une opération en O(1) à chaque fois. De plus, on fait **3*n** opérations en O(1) ; on a pas de boucle imbriquées. En conclusion, la complexété de cet algorithme est : **O(n)**

**6.b.** Quelle est la taille de la sortie comparée à la taille de l'entrée ?

**Réponse :**

**la taille de la sorie = la taille de l'entrée**

**Justification** : La taille de la sortie de la liste est égale à la taille de l'entrée de la lsite car par exemple, lorsqu'on met **3 éléments** dans chaque liste au départ, lorsqu'on fait la somme de la taille de chaque liste on obtient **9**. Or la sortie de la liste nous donne une liste de l'union des trois listes qui nous donne aussi **9**. Par conséquent, la taille de la sortie = la taille de l'entrée.

On veut montrer que les instances positives (respectivement négatives) de **TRILISTE** ont pour image des instances positives (respectivement négatives) de **ZERO**.

Autrement dit, on veut montrer l'équivalence suivante :

$$(X, Y, Z) \in TRILISTE \iff r(X, Y, Z) \in ZERO$$

## 2.4 Question 7.

**7.a.** Calculez les images par la transformation r des instances $(X_p, Y_p, Z_p)$ et $(X_n, Y_n, Z_n)$ données à la Question 4 et vérifiez que l'instance positive (respectivement négative) a une image qui est une instance positive (respectivement négative) de ZERO.

**Réponse :**

**Instance 1** :

$$\begin{cases} X_p = [2, -4, -7] \\ Y_p = [-5, 8, -6, -1] \\ Z_p = [-3, -10, 9] \end{cases}$$

Soit $\mathbf{r}(X_p, Y_p, Z_p)$ l'image par la transformation r des instances $(X_p, Y_p, Z_p)$, on a :

$\mathbf{r}(X_p, Y_p, Z_p)$ = **[21, -39, -69, -48, 82, -58, -8, -33, -103, 87]**.

$\mathbf{r}(X_p, Y_p, Z_p)$ est une instance positive de ZERO car $\forall x,\ y,\ z,\ x \in X_p, y \in Y_p$ et $z \in Z_p$,

**pour x = 21, y = 82 et z = -103, on a 21 + 82 - 103 = 0**.

**Instance 2** :

$$\begin{cases} X_n = [2, 4, 7] \\ Y_n = [5, 8, 6] \\ Z_n = [3, 10, 9] \end{cases}$$

Soit $\mathbf{r}(X_n, Y_n, Z_n)$ l'image par la transformation r des instances $(X_n, Y_n, Z_n)$, on a :

$\mathbf{r}(X_n, Y_n, Z_n)$ = **[21, 41 71, 52, 82, 62, 27, 97, 87]**.

$\mathbf{r}(X_n, Y_n, Z_n)$ est une instance négative de ZERO car $\forall x,\ y,\ z$, il n'existe pas de x, y, z $\in (X_n, Y_n, Z_n)$ tels que x + y + z = 0. De plus, tous les éléments de la liste sont positives donc on ne peut pas avoir une instance positive de ZERO.

**7.b.** Montrez que si (X, Y, Z) $\in$ TRILISTE alors r(X, Y, Z) $\in$ ZERO

**Réponse :**

D'après la définition du TRILISTE, on a :

(X, Y, Z) ∈ TRILISTE $\Longrightarrow$ ∃ x ∈ X, y ∈ Y et z ∈ Z | x + y + z = 0   ①

Soient $x_r, y_r, z_r \in r(X,Y,Z)$, on a :

$$\begin{cases} x_r = 10x + 1 & ② \\ y_r = 10y + 2 & ③ \\ z_r = 10z - 3 & ④ \end{cases}$$

En faisant la somme de ② + ③ + ④, on a :

$$x_r + y_r + z_r = 10x + 1 + 10y + 2 + 10z - 3$$

$$\Longrightarrow x_r + y_r + z_r = 10(x + y + z) + (1 + 2 - 3)$$

$$\Longrightarrow x_r + y_r + z_r = 10(x + y + z) + 0$$

or d'après ①, x + y + z = 0 donc $x_r + y_r + z_r = 0$   ⑤

Par conséquent, de ① et ⑤, on en déduit que si (X, Y, Z) ∈ TRILISTE alors r(X, Y, Z) ∈ ZERO

**7.c.** Montrez que si r(X, Y, Z) ∈ ZERO alors (X, Y, Z) ∈ TRILISTE.

**Réponse :**

Soit **r** la transformation des instances TRILISTE en instance ZERO :

si **r(X, Y, Z)** $\in$ ZERO, alors supposons qu'il existe un $\beta, \delta, \lambda \in \mathbb{R}$ avec $\beta \in \{1, 2, -3\}, \delta \in \{1, 2, -3\}, \lambda \in \{1, 2, -3\}$ et $x_r, y_r, z_r \in r(X, Y, Z)$ tel que :

$$\begin{cases} x_r = 10x + \beta & ① \\ y_r = 10y + \delta & ② \\ z_r = 10z + \lambda & ③ \end{cases}$$

r(X, Y, Z) $\in$ ZERO $\Longrightarrow x_r + y_r + z_r = 0$ ④

En faisant la somme de ① + ② + ③, on a :

$$x_r + y_r + z_r = 10\text{x} + \beta + 10\text{y} + \delta + 10\text{z} + \lambda$$
$$\Longrightarrow x_r + y_r + z_r = 10(\text{x} + \text{y} + \text{z}) + (\beta + \delta + \lambda)$$
$$④ \Longrightarrow 10(\text{x} + \text{y} + \text{z}) + (\beta + \delta + \lambda) = 0$$

$$\Longrightarrow \begin{cases} x + y + z = 0 \ et \ \beta + \delta + \lambda = 0 & ⑤ \\ 10(x + y + z) = -(\beta + \delta + \lambda) \neq 0 & ⑥ \end{cases}$$

$$⑤ \Longrightarrow \beta + \delta + \lambda = 0 \Longrightarrow \lambda = -(\beta + \delta)$$

Lorsqu'on prend $(\beta, \delta, \lambda) = (1, 2, -3)$, c'est la seule solution pour que $\beta + \delta + \lambda = 0$

$$\Longrightarrow \begin{cases} x_r = 10x + 1 \\ y_r = 10y + 2 \\ z_r = 10z - 3 \end{cases}$$

D'après ⑤, on vient de prouver que si r(X, Y, Z) $\in$ ZERO alors (X, Y, Z) $\in$ TRILISTE.
L'équation ⑥ n'a pas de solution, car quelque soit les valeurs de $\beta, \delta, \lambda$, $|\beta+\delta+\lambda| < 10$ et $|10(x + y + z)| \geq 10$.
Par conséquent, d'après ⑤ et ⑥, si r(X, Y, Z) $\in$ ZERO alors (X, Y, Z) $\in$ TRILISTE

**7.d.** Concluez.

**Réponse :**

En guise de conclusion, d'après les questions 7.b et 7.c, on a une équivalence entre les deux démonstrations :

$$\mathbf{(X, Y, Z) \in TRILISTE \iff r(X, Y, Z) \in ZERO}$$

## 2.5 Question 8.

**8.a.** Expliquez rigoureusement comment, grâce à cette transformation r qui vérifie l'équivalence (X, Y, Z) $\in$ TRILISTE $\iff$ r(X, Y, Z) $\in$ ZERO, on peut résoudre le problème TRILISTE à partir de la fonction testZERO qui résout le problème ZERO.

**Réponse :**

Tout d'abord, le problème de TRILISTE consiste à vérifier s'il existe trois éléments $x \in X$**,** $y \in Y$ **et** $z \in Z$ tel que $\mathbf{x} + \mathbf{y} + \mathbf{z} = \mathbf{0}$.

Ensuite, la fonction **testZERO** prend en entrée une liste et renvoie vraie ou faux si cette liste est une instance positive ou une instance négative de zéro.

En outre, la fonction **reduction** réduit le problème de TRILISTE à un problème classique de ZERO.

Lorsqu'on applique la fonction reduction sur le problème de TRILISTE, elle nous renvoie une liste et lorsqu'on applique la fonction testZERO sur cette liste elle nous envoie vraie ou faux si la liste est une instance positive de ZERO ou pas.

Enfin, d'après la question 7.c, on a : si r(X, Y, Z) $\in$ ZERO alors (X, Y, Z) $\in$ TRILISTE.

Par conséquent, on peut résoudre le problème TRILISTE à partir de la fonction testZERO qui résout le problème ZERO.

**8.b.** En déduire une fonction **testTRILISTE** qui détermine si (X, Y, Z) est une instance positive ou négative de TRILISTE et qui fait appel à la fonction reduction de la Question 5 et à la fonction testZERO de la Question 3.

**Réponse :**

```
def testTRILISTE(X, Y, Z):
    r = reduction(X, Y, Z) // Complexité en O(n)
    return testZERO(r)     // Complexité en O(n^2)

```

## 2.6 Question 9.

**Question :** Quelle est la complexité de la fonction testTRILISTE ? Justifiez soigneusement votre réponse.

**Réponse :**

La complexité de la fonction **reduction** est en O(n) ; ①

La complexité de la fonction **testZERO** est en O($n^2$) ; ②

De ① et ②, on en déduit que la complexité de la fonction **testTRILISTE** est en :

$$\mathbf{O}(n^2) + \mathbf{O(n)} \simeq \mathbf{O}(n^2)$$

## 2.7 Question 10.

**Question :** Le problème de décision TRILISTE est-il dans P ? dans NP ? dans EXP ?

**Réponse :**

Le problème de la décision de **TRILISTE** est dans **P** car sa complexité est **Polynomiale**.