

Concours blanc - Informatique

Lundi 2 juin 2025 - 13h30-15h30

L'utilisation des calculatrices n'est pas autorisée pendant cette épreuve.

On attachera une grande importance à la concision, à la clarté et à la précision de la rédaction. L'indentation devra être matérialisée par un trait vertical et les codes non triviaux devront impérativement être commentés.



Dans tout le sujet, on utilise le mot *entier* pour désigner un entier naturel.

Partition équilibrée

Ce sujet a pour objectif de résoudre de différentes manières le problème de *partition équilibrée*.

On considère un ensemble d'entiers $\{x_0, \dots, x_{n-1}\}$. Pour une partie $I \subset \llbracket 0; n-1 \rrbracket$, on note $J = \llbracket 0; n-1 \rrbracket \setminus I$ son complémentaire. On appelle *partition* de $\{x_0, \dots, x_{n-1}\}$ un couple $(\{x_i, i \in I\}, \{x_i, i \in J\})$ avec $I \subset \llbracket 0; n-1 \rrbracket$. On définit l'*écart* d'une partition $(\{x_i, i \in I\}, \{x_i, i \in J\})$ par

$$\left| \sum_{i \in I} x_i - \sum_{i \in J} x_i \right|$$

On note $\mathcal{P}(\llbracket 0; n-1 \rrbracket)$ l'ensemble des parties de $\llbracket 0; n-1 \rrbracket$. Le minimum des écarts de toutes les partitions

$$\min_{I \in \mathcal{P}(\llbracket 0; n-1 \rrbracket)} \left| \sum_{i \in I} x_i - \sum_{i \in J} x_i \right|$$

est appelé *écart minimal*. Enfin, on appelle *partition équilibrée* une partition dont l'écart est égal à l'écart minimal.

Dans ce qui suit, un ensemble sera représenté par une liste python.

Par exemple, avec l'ensemble $L = [3, 2, 1, 2]$, le couple $([3, 1], [2, 2])$ est une partition équilibrée car la quantité

$$0 = |(3 + 1) - (2 + 2)|$$

est l'écart minimal. On considère que les couples $([3, 1], [2, 2])$, $([1, 3], [2, 2])$, $([2, 2], [3, 1])$ et $([2, 2], [1, 3])$ forment une même partition équilibrée puisque seul l'ordre des listes ou des éléments au sein des listes diffère. Des couples qui forment une même partition sont dits *doublons*.

Avec l'ensemble $L = [1, 1, 2, 3]$, le couple $([1, 1, 2], [3])$ est une partition équilibrée car la quantité

$$1 = |(1 + 1 + 2) - 3|$$

est l'écart minimal.

Rappels

En langage python, on rappelle les instructions suivantes :

- la fonction `abs` calcule la valeur absolue ;
- `float('inf')` renvoie $+\infty$.

 L'utilisation des fonctions `min`, `sorted`, `sum` et de la méthode `sort` n'est pas autorisée.

I Préliminaires

1 . Déterminer, sans justifier, toutes les partitions équilibrées pour les ensembles suivants :

$$[1, 2, 3, 4] \quad \text{et} \quad [1, 2, 3, 4, 5]$$

2 . Écrire une fonction `somme(L)` d'argument `L` une liste d'entiers qui renvoie la somme des termes de la liste `L` que l'on notera $s(L)$.

3 . Écrire une fonction `ecart(L1,L2)` d'arguments `L1`, `L2` des listes d'entiers qui renvoie $|s(L1) - s(L2)|$.

Dans ce qui suit, on suppose disposer d'une fonction `extract_couple(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie la liste de toutes ses partitions.

Par exemple, l'appel `extract_couple([1,2,1])` renvoie

```
[ [], [1, 2, 1]], [[1], [2, 1]], [[2], [1, 1]], [[1, 2], [1]],  
  [[1], [1, 2]], [[1, 1], [2]], [[2, 1], [1]], [[1, 2, 1], []] ]
```

4 . Déterminer les résultats des appels `extract_couple([1,2])` et `extract_couple([1,1])`.

5 . Justifier que le résultat de l'appel `extract_couple(L)` avec `L` une liste non vide d'entiers contient systématiquement des couples doublons.

II Partitions équilibrées

6 . Compléter la fonction `min_ecart(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie l'écart minimal de cet ensemble.

```
def min_ecart(L):  
    L_ecart=extract_couple(L)  
    res=float('inf')  
    for L1,L2 in L_ecart:  
  
        ...  
  
    return res
```

7 . Écrire une fonction `argmin_ecart(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie la liste de toutes les partitions équilibrées de cet ensemble. La fonction `argmin_ecart` utilisera la fonction `extract_couple` ainsi que la fonction `min_ecart` pour le calcul de l'écart minimal.

8 . Écrire une fonction `min_argmin_ecart(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie un couple formé de l'écart minimal et de la liste de toutes les partitions équilibrées de cet ensemble. L'utilisation de la fonction `extract_couple` est impérative et l'utilisation de fonction `min_ecart` est interdite. Le corps de la fonction `min_argmin_ecart` ne doit contenir qu'une seule boucle `for`.

III Représentation binaire

9 . Écrire une fonction `binaire(a,n)` d'arguments `a, n` des entiers avec $a \in \llbracket 0; 2^n - 1 \rrbracket$ qui renvoie l'écriture binaire de taille n (non nul) de l'entier $a = \sum_{i=0}^{n-1} d_i 2^i$ avec les $d_i \in \{0, 1\}$ sous forme de liste de booléens $[b_0, \dots, b_{n-1}]$ avec $b_i = \text{True}$ si $d_i = 1$ et $b_i = \text{False}$ sinon.

Par exemple, l'appel `binaire(1,5)` renvoie `[True, False, False, False, False]` et l'appel `binaire(2,5)` renvoie `[False, True, False, False, False]`.

10 . Écrire une fonction non récursive `liste_bin(n)` d'argument `n` entier qui renvoie la liste de tous les mots binaires de taille n sous forme de liste de booléens conformément au format décrit à la question précédente.

Par exemple, l'appel `liste_bin(2)` renvoie `[[False, False], [True, False], [False, True], [True, True]]`. L'utilisation de la fonction `binaire` est impérative.

11 . Écrire une fonction `part(L,ind)` d'arguments `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ et `ind` une liste de booléens $[b_0, \dots, b_{n-1}]$ qui renvoie le couple de listes correspondant aux ensembles $\{x_i, i \in \llbracket 0; n-1 \rrbracket \mid b_i = \text{True}\}$ et $\{x_i, i \in \llbracket 0; n-1 \rrbracket \mid b_i = \text{False}\}$.

12 . Écrire une fonction `extract_couple` qui soit conforme à la description faite dans la partie I. L'utilisation des fonctions `liste_bin` et `part` est impérative.

13 . Écrire une fonction récursive `liste_bin_rec(n)` d'argument `n` entier qui renvoie la liste de tous les mots binaires de taille n sous forme de liste de booléens conformément au format décrit à la question 10.

IV Partitions équilibrées par indexation

Dans cette partie, on se propose de réécrire des fonctions analogues à celles de la partie II sans utiliser le résultat de la fonction `extract_couple`.

14 . Écrire une fonction `somme_ind(L,ind)` d'arguments `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ et `ind` une liste de booléens $[b_0, \dots, b_{n-1}]$ qui renvoie le couple $\left(\sum_{i=0}^{n-1} x_i \delta_{b_i}, \sum_{i=0}^{n-1} x_i (1 - \delta_{b_i}) \right)$ avec $\delta_{b_i} = 1$ si $b_i = \text{True}$ et $\delta_{b_i} = 0$ sinon.

15 . Écrire une fonction `ecart_ind(L,ind)` d'arguments `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ et `ind` une liste de booléens $[b_0, \dots, b_{n-1}]$ qui renvoie

$$\left| \sum_{i=0}^{n-1} x_i \delta_{b_i} - \sum_{i=0}^{n-1} x_i (1 - \delta_{b_i}) \right|$$

16 . Écrire une fonction `min_ecart_ind(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie l'écart minimal de cet ensemble. L'utilisation des fonctions `liste_bin` et `ecart_ind` est impérative.

17 . Écrire une fonction `argmin_ecart_ind(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie la liste de toutes les partitions équilibrées de cet ensemble. L'utilisation des fonctions `liste_bin`, `min_ecart_ind`, `ecart_ind` et `part` est impérative.

V Suppression des doublons

18 . Écrire une fonction `tri(T)` d'argument `T` une liste de n entiers qui renvoie une nouvelle liste contenant les valeurs de `T` triées par ordre croissant. L'algorithme de tri implémenté doit être performant.

19 . Écrire une fonction `equal_part(P1,P2)` d'arguments `P1`, `P2` deux listes de partitions d'un ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie `True` si les partitions sont des doublons et `False` sinon.

Par exemple, l'appel `equal_part([[1,3],[2,2]],[[2,2],[3,1]])` renvoie `True`.

20 . Écrire une fonction `detect(P,R)` d'arguments `P` une partition et `R` une liste de partitions d'une ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie `True` si la partition `P` possède un doublon dans `R` et `False` sinon.

21 . Écrire une fonction `part_equi(L)` d'argument `L` une liste d'entiers correspondant à l'ensemble $\{x_0, \dots, x_{n-1}\}$ qui renvoie la liste de toutes les partitions équilibrées de cet ensemble sans doublon. L'utilisation des fonctions `argmin_ecart` (ou `argmin_ecart_ind`) et `detect` est impérative.