Exercice 5 : Partitions

12 juin

On considère le problème k-partition :

— Instance : un multi-ensemble S d'entiers de cardinal $n = k \times m$.

— Question : existe-t-il une partition de S en m sous-ensembles de même cardinal k et de même somme ?

Par exemple, le multi-ensemble $S = \{4, 5, 5, 5, 5, 6\}$ est une instance positive de 3-partition, car il existe une partition de S en deux triplets $S_1 = \{4, 5, 6\}$ et $S_2 = \{5, 5, 5\}$, de même somme 15 .

On s'intéresse dans un premier temps au problème 2-partition.

1. Écrire une fonction `deux_partition : int array -> bool` qui prend en argument un tableau d'entiers, qu'on supposera de taille paire $n = 2m$, représentant un multi-ensemble S et renvoie un booléen qui vaut true si et seulement si S est une instance positive de 2-partition. La fonction devra avoir une complexité en $\mathcal{O}(n \log n)$ et pourra modifier le tableau si nécessaire. On rappelle qu'on peut trier un tableau tab par la commande Array.sort compare tab.

On souhaite écrire une fonction pour résoudre le problème 3-partition. On représente une partition en triplets d'un tableau de taille $n = 3m$ par une permutation σ de $[\![0, n]\!]$ telle que le premier triplet est celui constitué des éléments d'indices $(\sigma(0), \sigma(1), \sigma(2))$, le deuxième triplet est constitué des éléments d'indices $(\sigma(3), \sigma(4), \sigma(5))$, etc.

On note que cette représentation n'est pas bijective, car deux permutations peuvent correspondre à la même partition.

2. Écrire une fonction `valide : int array -> int array -> bool` qui prend en argument un tableau sigma de taille n représentant une permutation σ de $[\![0, n]\!]$ et renvoie un booléen qui vaut true si cette permutation représente une partition en triplets qui sont tous de même somme dans le tableau à partitionner, et false sinon.

Pour résoudre le problème 3-partition, on souhaite énumérer les permutations de $\{0, \ldots, n-1\}$ dans l'ordre lexicographique. Par exemple, les permutations de $\{0, 1, 2\}$ dans l'ordre lexicographique sont $(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1)$ et $(2, 1, 0)$. On se donne une fonction `suivante : int array -> bool` qui prend en argument un tableau sigma de taille n représentant une permutation σ de $\{0, \ldots, n\}$ et :

— si σ n'est pas la dernière permutation selon l'ordre lexicographique, alors suivante sigma modifie sigma pour qu'il devienne la permutation suivante, et renvoie le booléen true ;

— si σ est la dernière permutation, alors suivante sigma renvoie false sans modifier le tableau.

4. Écrire une fonction `trois_partition : int array -> bool` qui prend en argument un tableau d'entiers, qu'on supposera de taille paire $n = 3m$, représentant un multi-ensemble S et renvoie un booléen qui vaut true si et seulement si S est une instance positive de 3-partition.
5. Quelle est la complexité temporelle de la fonction précédente ?

On considère le problème $ABC-$ partition :

— Instance : trois multi-ensembles A, B et C d'entiers, de même cardinal m.

— Question : existe-t-il une partition de $A \sqcup B \sqcup C$ en m sous-ensembles de même somme, chaque sous-ensemble contenant exactement un élément de A, un élément de B et un élément de C ?

On admet que ce problème est NP-complet.

6. En considérant un multi-ensemble $S = \{1000a+100 \mid a \in A\} \sqcup \{1000b+10 \mid b \in B\} \sqcup \{1000c+1 \mid c \in C\}$, montrer que 3-partition est NP-complet par une réduction depuis ABC-partition.