# Contrôle continu – mercredi 10 novembre 2021

*L'épreuve dure 1 heure. Les réponses doivent être justifiées et rédigées correctement. Toute question non résolue peut être admise dans la suite. Le barême fourni est indicatif et susceptible de modification.*

**(7 pts) Exercice 1.** *Pile et multi-dépilement*

La structure de données PILE possède classiquement trois opérations : EMPILER$(P, x)$ empile un élément $x$ sur la pile $P$, DÉPILER$(P)$ supprime et renvoie le sommet de la pile $P$, et ESTVIDE$(P)$ teste si la pile est vide.
On étend la structure de donnée avec une opération MULTIDÉPILER$(P, k)$ qui dépile $k$ éléments de $P$ et renvoie le $k^{\text{ème}}$ élément dépilé. Si la pile contient moins de $k$ éléments, elle est vidée et son dernier élément est renvoyé.
*On suppose que les trois opérations de* PILE *ont une complexité (pire cas)* $O(1)$.

(3 pts) **1. i.** Écrire l'algorithme MULTIDÉPILER.
**ii.** Quelle est sa complexité ?

(4 pts) **2.** On souhaite maintenant analyser le coût d'un nombre quelconque d'opérations EMPILER, DÉPILER et MULTIDÉPILER. On suppose qu'on effectue $n$ opérations, et que la pile est initialement vide.

**i.** Effectuer une analyse *pire cas* de la complexité totale des $n$ opérations.
**ii.** Montrer que le coût global des $n$ opérations est borné par $O(n)$.
**iii.** Quel est le coût amorti par opération ?

**(13 pts) Exercice 2.** *Empaquetage*

On a $n$ objets de tailles $T_{[0]}, \ldots, T_{[n-1]}$ telles que $0 < T_{[i]} < 1$ pour tout $i$, et $m$ boîtes de taille 1. On veut ranger les $n$ objets dans les $m$ boîtes : une boîte peut contenir plusieurs objets, mais la somme des tailles des objets contenus dans une boîte ne doit pas dépasser 1.
**Exemple.** Supposons qu'on a 7 objets, de tailles 0.2, 0.5, 0.4, 0.7, 0.1, 0.3 et 0.6, et 3 boîtes. Alors une solution est $\{0.2, 0.7\}$, $\{0.5, 0.4, 0.1\}$ et $\{0.3, 0.6\}$. En revanche, si on n'a que 2 boîtes, il n'y a pas de solution.

(1,5 pts) **1. i.** Pourquoi dans l'exemple fourni, il est impossible de ranger tous les objets dans 2 boîtes de taille 1 ?
**ii.** Donner une condition nécessaire sur $m$ en fonction des tailles des objets, pour qu'il y ait une solution.
**iii.** En considérant trois objets, tous de taille 0.6, montrer que la condition n'est pas suffisante.

On résout le problème par recherche exhaustive. Une *tentative* consiste à placer chaque objet dans une boîte. Une tentative est *valide* si aucune boîte ne contient des objets dont la somme des tailles dépasse 1. On numérote les boîtes de 0 à $m-1$. Une tentative est représentée par un $n$-uplets $S$ d'entiers entre 0 et $m-1$, tel que $S_{[i]} = j$ si l'objet $i$ va dans la boîte $j$.

(3 pts) **2. i.** Écrire un algorithme ESTVALIDE$(T, S)$ qui teste si $S$ est une tentative valide.
**ii.** Quelle est la complexité de l'algorithme ESTVALIDE ?

(4,5 pts) **3.** On veut parcourir toutes les tentatives possibles, c'est-à-dire tous les $n$-uplets $S$ possibles.

**i.** Combien y a-t-il de $n$-uplets $S$ à parcourir ? Dans l'ordre lexicographique (vu en cours), quel est le premier $n$-uplet ? Et le dernier ?
**ii.** Écrire un algorithme SUIVANT qui prend en entrée un $n$-uplet $S$ et $m$, et renvoie le $n$-uplet suivant dans l'ordre lexicographique. La fonction renvoie « Fini » si $S$ est le dernier $n$-uplet.
**iii.** Quelle est la complexité (pire cas) de l'algorithme SUIVANT ?

(4 pts) **4. i.** Déduire des questions précédentes un algorithme EMPAQUETAGE qui prend en entrée un tableau $T$ de taille $n$ et un entier $m$ et renvoie VRAI si on peut empaqueter les $n$ objets de tailles $T_{[0]}, \ldots, T_{[n-1]}$ dans $m$ boîtes de taille 1.
**ii.** Analyser la complexité de l'algorithme EMPAQUETAGE.
**iii.** Supposons qu'on n'ait plus $m$ en entrée. Décrire un algorithme pour trouver le nombre minimal $m$ de boîtes nécessaires pour empaqueter les $n$ objets.

**5.** *(bonus)* Écrire une version *backtrack* de l'algorithme EMPAQUETAGE.