

## Algorithmique

8 Novembre 2010

- Le seul document autorisé est une feuille A4 recto-verso.
- Les calculatrices, téléphones, ordinateurs etc... sont interdits.

**Nom :**

**Prénom :**

**Section :**

| Exercice 1  | Exercice 2  | Exercice 3  | Exercice 4  |
|-------------|-------------|-------------|-------------|
| / 10 points | / 15 points | / 25 points | / 15 points |
|             |             |             |             |

|                   |
|-------------------|
| <b>Total / 65</b> |
|                   |

**Problème 1 [10 points].**

Soit  $A(n) = \sum_{k=1}^n k \cdot k!$ .

1. Calculer  $A(n)$  pour  $n = 1, 2, 3, 4, 5$ .
2. Comparer  $A(n)$  et  $n!$  pour  $n = 1, 2, 3, 4, 5$ .
3. Deviner une formule générale pour  $A(n)$  et prouver cette formule par induction.





**Problème 2 [15 points].** Etant donné  $S \subseteq \mathbb{N}$  et  $x \in \mathbb{N}$ , on considère le problème suivant : est-ce qu'il existe un ensemble  $T \subseteq S$  tel que  $x = \sum_{i \in T} i$  ? (Ce problème est appelé le problème du *subset sum*).

1. Donner la spécification formelle de ce problème.
2. On suppose que  $S$  est donné par  $\{s_1, s_2, \dots, s_n\}$  satisfaisant la propriété suivante :  $s_2 > s_1$ ,  $s_3 > s_2 + s_1$ , et en général  $s_k > \sum_{i=1}^{k-1} s_i$  pour tout  $k$ . Donner un algorithme qui trouve la réponse en  $n$  additions au plus. Expliquer pourquoi l'algorithme marche.





**Problème 3 [25 points].**

1. Supposons que nous disposons uniquement de la structure de données stack. Est-il possible de réaliser une file d'attente en utilisant deux stacks ? Si oui, donner les algorithmes correspondants pour les opérations de la file d'attente.
2. Nous voulons de nouveau réaliser une file d'attente en utilisant deux stacks, mais avec la condition de plus que le temps de calcul des opérations est similaire à celui d'une implémentation classique de file d'attente.

Plus précisément, supposons que les opérations sur les stacks se font en temps  $O(1)$  (i.e. ne dépendent pas de la taille du stack). Donner alors des algorithmes pour les opérations d'une file d'attente basée sur deux stacks qui ont la propriété que, si l'on commence avec une file vide, pour faire les  $m$  premières opérations de la file d'attente, on a besoin de  $O(m)$  opérations du stack.







**Problème 4 [15 points].**

Soit  $T$  un arbre binaire AVL de recherche, de hauteur  $h$  (on rappelle que  $\text{bal}(k)$  est défini comme la hauteur du sous-arbre de gauche de  $k$  moins la hauteur du sous-arbre de droite de  $k$ ; dans un arbre AVL, pour tout sommet  $k$ ,  $\text{bal}(k) \in \{0, +1, -1\}$ ). Un élément est inséré dans  $T$ , et un arbre binaire de recherche  $T'$  est obtenu. Répondre aux questions suivantes concernant  $T'$  *avant qu'aucune rotation soit effectuée*.

1. Quel est le *plus petit* nombre possible de sommets avec  $\text{bal}(k) \in \{+2, -2\}$ ? Justifier brièvement la réponse.
2. Quel est le *plus grand* nombre possible de sommets avec  $\text{bal}(k) \in \{+2, -2\}$ ? Montrer un exemple avec ce nombre et expliquer pourquoi c'est le plus grand nombre possible.
3. Soit  $s$  un sommet de  $T'$ , et soit  $t$  son fils de gauche. Est-il possible d'avoir  $\text{bal}(t) = +2$  et  $\text{bal}(s) = -1$ ? Si oui, donner un exemple. Si non, expliquer pourquoi.



