

Programmation avancée

Examen écrit, première session, 9 janvier 2018

Livres fermés. Durée : 3h30.

Remarques

- Répondez à chaque question sur une feuille **séparée**, sur laquelle figurent votre nom.
- Soyez bref et concis, mais précis.
- Sauf mention explicite, toutes les complexités sont à décrire par rapport au temps d'exécution des opérations concernées. Soyez toujours le plus précis possible dans le choix de votre notation (Ω , O ou Θ).

Question 1 : Vrai ou faux

Pour chacune des affirmations suivantes, déterminez si elle est vraie ou fausse et justifiez brièvement votre choix :

1. Soit deux algorithmes récursifs dont les complexités sont données respectivement par :
 - $T_1(n) = 2 * T_1(n/2) + 1$,
 - $T_2(n) = 3 * T_2(n/4) + n$.On a $T_1(n) \in O(T_2(n))$.
2. Soit l'encodage d'un arbre binaire complet sous la forme du tableau ci-dessous :

[4, 2, 6, 1, 5, 7, 9, 0].

Cet arbre satisfait à la propriété d'arbre binaire de recherche.

3. L'implémentation par un tas max d'une file à priorité est la plus adaptée lorsque le nombre d'insertions est nettement plus élevé que le nombre d'extractions du maximum.
4. Soit la fonction de hachage, $h(x) = x \bmod m$. L'insertion des clés suivantes dans cet ordre :

$0, 1, 2, \dots, m/2 - 2, m/2 - 1, m/2, \dots, m, m + 1, m + 2, \dots, 3m/2 - 2, 3m/2 - 1, 3m/2$

sera plus efficace avec le sondage linéaire qu'avec le chaînage.

5. L'ordre d'affichage des clés lors du parcours préfixe d'un arbre binaire de recherche n'est pas impacté par l'ordre dans lequel ces clés ont été insérées dans l'arbre.

Question 2 : Analyse d'algorithmes

Soit l'algorithme de tri récursif suivant :

- Si le tableau a une taille inférieure ou égale à 1, ne rien faire.
- Si l'élément le plus à gauche du tableau est plus large que l'élément le plus à droite, les échanger
- S'il y a au moins trois éléments dans le tableau, (i) on trie les premiers deux tiers¹ du tableau récursivement, (ii) on trie les derniers deux tiers du tableau, (iii) on retri les premiers deux tiers à nouveau.

1. Prouvez que cet algorithme est correct.
2. Est-il stable, en place ?
3. Analyser sa complexité en temps et en espace. Justifiez votre réponse. Si vous utilisez un théorème, énoncez-le.

Question 3 : Tri

1. Décrivez l'algorithme de tri rapide (sans randomization du pivot) dans le formalisme de votre choix. Donnez sa complexité dans le meilleur et le pire cas.
2. Soit le tableau $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, permutez les éléments de ce tableau de manière à ce que l'exécution de votre tri rapide sur ce tableau corresponde au meilleur cas, c'est-à-dire avec la propriété que toutes les partitions produiront des sous-tableaux ne différant pas plus d'un élément en taille.
3. Implémentez une fonction $\text{BESTQSARRAY}(N)$ générant un tableau contenant N clés distinctes (par exemple les entiers de 1 à N) correspondant au meilleur cas pour le tri rapide.
4. Analysez la complexité en temps de votre fonction BESTQSARRAY .

Question 4 : Structures de données

1. Qu'est-ce qu'un tas-max ?
2. Décrivez dans le formalisme de votre choix la fonction HEAP-INSERT insérant une clé dans un tas-max et énoncez sa complexité dans le pire et le meilleur cas.
3. Donnez le pseudo-code d'un algorithme $\text{LARGESTK}(Q, k, x)$ déterminant si la k ème plus grande valeur du tas Q est plus grande ou égale à une valeur x (ou de manière équivalente, déterminant s'il y a au moins k valeurs dans le tas supérieures ou égales à x). L'algorithme doit être linéaire par rapport à k et constant par rapport à la taille du tas.

1. Si la taille du tableau n'est pas divisible par 3, les tailles des sous-tableaux sont arrondies vers le haut.

Question 5 : résolution de problème

Vous partez au ski et vous souhaitez louer votre équipement. La boutique de location propose trois prix :

- Un prix à la journée $f_1 = 25\text{€}$,
- Un prix pour deux jours $f_2 = 40\text{€}$,
- Un prix pour quatre jours $f_4 = 60\text{€}$.

Vous souhaitez passer N jours à la station de ski et vous disposez, en début de séjour, des prévisions météo vous permettant de déterminer pour chaque jour i ($i = 1, \dots, N$) si le temps sera beau ou pas. Cette information est encodée dans une table $B[1..N]$ tel que $B[i] = 1$ si le temps sera beau au jour i et $B[i] = 0$ sinon. La table ci-dessous donne un exemple de prévisions pour $N = 10$ jours :

j	1	2	3	4	5	6	7	8	9	10
$B[j]$	1	0	1	1	0	1	1	0	1	1

Sachant que vous souhaitez skier tous les jours de beau temps et seulement ces jours-là, le problème est de déterminer quand et pour combien de temps louer votre équipement de manière à minimiser vos dépenses. Par exemple, une location à la journée pour l'exemple de la table ci-dessus reviendrait à 175€ (7 journées de beau temps à 25€ chaque).

1. Une approche pour résoudre le problème pourrait être la suivante : Choisir la durée de location le premier jour de la manière suivante :
 - S'il fait mauvais le premier jour, ne rien prendre.
 - Sinon, s'il reste au moins 4 jours et s'il y a plus de 3 jours de beau temps parmi les 4 premiers jours, prendre un pack de 4 jours.
 - Sinon, s'il reste au moins deux jours et qu'il fait beau les deux premiers jours, prendre un pack 2 jours
 - Sinon, prendre un prix à la journée

Une fois la première location expirée, utiliser la même stratégie pour les jours restants.

- (a) Sur quel type de résolution de problème est basée cette solution ?
 - (b) Quelle serait la complexité de l'algorithme résultant, en fonction du nombre de jours N
 - (c) Que donnera cet algorithme sur le problème de la table ci-dessus ?
 - (d) Prouvez que cet algorithme n'est pas correct.
2. On aimerait mettre au point un algorithme optimal pour résoudre le problème en utilisant la programmation dynamique. On se propose pour cela de calculer le coût $C(j)$ défini, pour $j = 1, \dots, N$, comme le prix minimum de la location entre les jours j et N inclus (en supposant qu'on arrive au jour j).
 - (a) Pour quelle valeur de j , $C(j)$ est la solution recherchée ?
 - (b) Formulez $C(j)$ de manière récursive. Précisez bien le cas de base.
 - (c) En déduire le pseudo-code d'un algorithme efficace pour résoudre le problème.
 - (d) Calculez les valeurs $C(j)$ pour la table donnée en exemple.