

Contrôle continu

La durée du contrôle continu est 1h30. Il est constitué de trois exercices indépendants. Toute question non résolue peut être admise dans la suite. Les réponses doivent être correctement rédigées. Aucun document n'est autorisé. Le barème, indicatif, est sur 22 points.

Exercice 1 (sur 6 pts).

Du cours

1. Soit $T_1 = [9, 7, 8, 4, 1, 3, 5, 6, 2]$ et $T_2 = [9, 8, 3, 6, 7, 1, 2, 5, 4]$.

i. L'un des deux n'est pas un tas, lequel ? Justifier.

On note T le tableau qui est un tas parmi les deux. On voit T comme une liste de priorité : une entrée x est un élément de valeur x et de priorité x .

ii. Appliquer l'algorithme EXTRAIRE pour supprimer et renvoyer l'élément le plus prioritaire sur T . Donner l'état de T après l'algorithme, sous les deux formes : un tableau et un arbre quasi-complet.

2. i. Dessiner l'arbre binaire de recherche obtenu en insérant successivement (dans cet ordre) les entiers 4, 1, 7, 8, 9, 3 et 2.

ii. Dessiner un arbre binaire de recherche contenant exactement les mêmes entiers, de hauteur la plus petite possible. Justifier qu'il n'en existe pas de hauteur inférieure.

3. Qu'est-ce qu'une « collision » dans une table de hachage ?

Exercice 2 (sur 6 pts).

Tableau circulaire croissant

Un *tableau circulaire croissant d'origine k* est un tableau T de n entiers distincts tel que $T_{[k]} < T_{[k+1]} < \dots < T_{[k+n-1]}$ où les indices sont pris modulo n . On étudie le problème suivant : étant donné un tableau circulaire croissant T , trouver son origine k .

Exemple : *tableau circulaire croissant d'origine $k = 10$.*

9	13	16	18	19	23	28	31	37	42	1	3	4	5	7	8
---	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---

1. Écrire un algorithme de complexité linéaire pour le problème.
2.
 - i. Écrire un algorithme de type « diviser pour régner » pour le problème.
 - ii. Analyser la complexité de l'algorithme.
3. Décrire un algorithme qui prend en entrée un tableau circulaire croissant et un entier x , et renvoie VRAI si x appartient à T et FAUX sinon. Une description en une ou deux phrases suffit (pas besoin de pseudo-code). Analyser la complexité de votre algorithme, qui doit être le plus efficace possible.

Exercice 3 (sur 10 pts).*Compteur avec remise à zéro*

On considère un compteur binaire, vu comme un tableau C de k bits. La valeur du compteur C est l'entier $v = \sum_{i=0}^{k-1} C_{[i]} 2^i$. Le compteur possède deux opérations :

- INCRÉMENT(C) ajoute 1 au compteur *modulo* 2^k : si la valeur est $2^k - 1$, elle passe à 0 et il n'y a pas d'erreur :
- ZÉRO(C) remet le compteur à 0 (tous ses bits sont à zéro).

Exemple : la valeur de $C = [1, 1, 0, 1, 0]$ est 11, qui s'écrit 1011 en binaire.

1. Quel est le coût, dans le pire cas, d'un appel à INCRÉMENT ou à ZÉRO ?

On étudie la complexité amortie de ces opérations. On définit le potentiel $\Phi(C)$ d'un tableau C comme le nombre de 1 qu'il contient. *Dans l'exemple, $\Phi(C) = 3$.*

2. On applique une opération INCRÉMENT ou ZÉRO à un compteur C . On note Φ_0 le potentiel avant l'opération, Φ_1 le potentiel après l'opération, et c le nombre de cases de C modifiées par l'opération.
 - i. Montrer que dans le cas d'INCRÉMENT, $c + \Phi_1 - \Phi_0 \leq 2$.
 - ii. Montrer que dans le cas de ZÉRO, $c + \Phi_1 - \Phi_0 = 0$.
 - iii. En déduire que le coût amorti d'une suite de n INCRÉMENTS en partant d'un compteur initialisé à 0 est $O(1)$ par opérations, si le coût est compté en nombre d'écriture dans des cases.

L'analyse précédente est trop simpliste car on ne compte que les écritures dans C . Pour ZÉRO, il faut aller lire chaque case pour savoir si elle est à modifier ou non : on ne peut pas considérer ces lectures comme gratuites.

On ajoute au compteur un indice $i_{\max} = \max\{i : C_{[i]} = 1\}$, avec $i_{\max} = -1$ si $C_{[i]} = 0$ pour tout i . Le compteur est maintenant le couple $(C, i_{\max})$ et l'algorithme ZÉRO devient : « pour $i = 0$ à $i_{\max} : C_{[i]} \leftarrow 0 ; i_{\max} \leftarrow -1$ ».

3.
 - i. En notant c le coût de l'opération ZÉRO, Φ_0 le potentiel de C avant ZÉRO et Φ_1 son potentiel après, montrer que $c + \Phi_1 - \Phi_0$ n'est pas borné par une constante.
 - ii. On définit la nouvelle fonction potentiel $\Psi(C, i_{\max}) = \Phi(C) + i_{\max}$. On applique l'opération ZÉRO, et on note c son coût, Ψ_0 et Ψ_1 les potentiels avant et après l'opération. Montrer que cette fois-ci, $c + \Psi_1 - \Psi_0 \leq 1$.
 - iii. Justifier qu'avec ce nouveau compteur, une suite de n opérations INCRÉMENT et ZÉRO a un coût amorti par opérations $O(1)$.