

Structures de données et algorithmes

Examen écrit, 14 juin 2012

Livres fermés. Durée maximale : 3h30.

Remarques

- Répondez à chaque question sur une feuille séparée, sur laquelle figurent votre nom et votre section. 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) et justifiez votre réponse.

Question 1

L'algorithme MATCH reçoit en entrée un tableau $S[1..n]$ et un pattern $P[1..m]$ où $1 \leq m \leq n$ et renvoie la position l de la première occurrence du pattern P dans le tableau S , c'est-à-dire $l = p$ si $S[p..p+m-1] = P$ et $l = -1$ si le pattern P n'apparaît pas dans le tableau S .

```
MATCH( $P, S, n, m$ )
1   $l = 0$ 
2   $matched = \text{FALSE}$ 
3  while ( $l \leq n$ )  $\wedge$   $\neg matched$ 
4       $l = l + 1$ 
5       $r = 1$ 
6       $matched = \text{TRUE}$ 
7      while ( $r \leq m$ )  $\wedge$   $matched$ 
8           $matched = matched \wedge (P[r] == S[l + r - 1])$ 
9           $r = r + 1$ 
10 if  $matched$ 
11     return  $l$ 
12 else return  $-1$ 
```

- L'algorithme tel que présenté est incorrect. Repérez l'erreur et corrigez la.
- Calculez la complexité au pire cas de l'algorithme corrigé en fonction de m et n et donnez un exemple correspondant à ce pire cas.
- Calculez la complexité au meilleur cas de l'algorithme corrigé en fonction de m et n et donnez un exemple correspondant à ce meilleur cas.
- Modifiez cet algorithme de manière à ce qu'il fournisse le nombre d'occurrences du pattern P dans le tableau S à la place de la position du premier pattern (ces occurrences peuvent se chevaucher).
- Donnez les complexités au meilleur et au pire cas de l'algorithme proposé au point (d).

Question 2

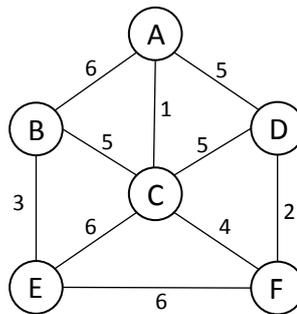
- Expliquez le principe d'une pile et d'une file et décrivez pour chacune de ces structures une manière de l'implémenter en précisant la complexité des opérations d'insertion et de retrait d'éléments dans la structure.
- Montrez comment implémenter une file à l'aide de deux piles et analysez la complexité des différentes opérations.

Question 3

- Illustrez l'arbre binaire de recherche dont le parcours préfixe donne :
10 5 2 - - 7 - - 30 20 15 - - - -
Note : un - correspond à l'absence d'un sous-arbre.
- Déterminez une séquence d'opérations d'insertion permettant d'obtenir l'arbre du point (a).
- Précisez les rotations, en nombre minimal, à appliquer à l'arbre du point (a) pour le rendre AVL.
- Décrivez un algorithme de complexité $\Theta(n)$ permettant de parcourir selon l'ordre infixe un arbre binaire de recherche composé de n noeuds.
- Montrez que la complexité de cet algorithme est $\Theta(n)$.

Question 4

- Qu'est-ce qu'un arbre couvrant ? Un arbre couvrant de poids minimum ? Donnez un exemple d'application.
- Décrivez l'algorithme de Kruskal pour la recherche d'un arbre couvrant de poids minimum, dans le formalisme de votre choix.
- Illustrez graphiquement chaque étape de l'exécution de l'algorithme sur le graphe ci-dessous.



- Que retourne l'algorithme de Kruskal lorsqu'on l'applique à un graphe non connexe ?

Question 5

On cherche à écrire une fonction `LONGEST-CST-SUBARRAY` qui renvoie les bornes d'une plus longue suite de valeurs constantes dans un tableau $A[1..n]$. Si $(i, j) = \text{LONGEST-CST-SUBARRAY}(A)$, $A[i..j]$ est une plus longue suite de valeurs constantes de A ($A[i] = A[i+1] = \dots = A[j]$).

- Expliquez le principe d'une approche force-brute pour résoudre ce problème et donnez sa complexité.
- Ecrivez une solution de type diviser-pour-régner et calculez sa complexité.
- Ecrivez une solution par programmation dynamique et calculez sa complexité.