Université Lille1 Sciences et Technologies

Info 204 - ASD

2013–2014
Licence STS - Semestre 4

DS2 - documents de cours, TD, TP autorisés - 2h

A lire impérativement avant de commencer

Vous pourrez utiliser toute fonction vue en cours, TD, TP à la condition expresse de spécifier correctement les entrées et les sorties.

Les temps indiqués pour chaque exercice sont approximatifs et seulement là pour vous aider à répartir votre temps de travail.

Exercice 1 : Bord d'un mot - 30 minutes - 5 points

On rappelle la définition de bord d'un mot. Soit un alphabet Σ et u un mot sur Σ^+, on appelle *bord* de u un mot $v \in \Sigma^\star$ tel que v est à la fois suffixe et préfixe du mot u et $v \neq u$. On notera `Bord`(u) la longueur du plus long mot bord de u.

Q 1.1 Donnez le plus long bord des mots suivants : `a`, `at`, `ata`, `atac`, `ataca`, `atacat`, `atacata`, `atacatac`.

Q 1.2 Proposez une fonction de calcul de `Bord`(u) en OCaml qui prend en entrée une chaîne de caractères et retourne la longueur du plus long bord.

Q 1.3 Quelle est la complexité en temps et en espace de votre algorithme ? Justifiez.

Exercice 2 : AVL - 30 minutes - 5 points

On considère ici des AVL d'entiers déclarés ainsi :

```
type avl = Vide | Cons of noeud
and noeud = { valeur : int; mutable droit : avl; mutable gauche : avl }
```

Q 2.1 Dessiner l'AVL obtenu après chaque insertion des valeurs suivantes, dans cet ordre : 8, 5, 4, 1, 3, 9, 10. Pour chaque valeur insérée vous indiquerez le nombre de rotations qui ont été nécessaires.

Q 2.2 Ecrire le code d'une fonction `desequilibre : avl -> int` qui calcule la valeur de déséquilibre du nœud racine de l'arbre passé en paramètre.

Q 2.3 Quel est le comportement asymptotique en temps de cette fonction ? Justifiez.

Exercice 3 : ABR et tas - 15 minutes - 3 points

Un tas est nécessairement un arbre binaire quasi-parfait. Mais est-il toujours possible d'organiser un ensemble de n valeurs (n quelconque) en tas max de manière à ce que cet arbre binaire soit aussi un arbre binaire de recherche ?

Q 3.1 Est-ce vrai lorsque $n = 1$? Justifiez.

Q 3.2 Est-ce vrai lorsque $n = 2$? Justifiez.

Q 3.3 Est-ce vrai lorsque $n \geq 3$? Justifiez par un raisonnement ou grâce à un contre-exemple.

Exercice 4 : Choix d'une structure de données - 45 minutes - 7 points

Dans cet exercice on considère un algorithme de tri d'une liste qui utilise une structure de données annexe s possédant les primitives suivantes :

- `nouvelle ()` qui permet de créer une structure s vide,
- `inserer s e` qui permet d'insérer un entier e dans la structure s,
- `extraireMax s` qui retourne la valeur maximale stockée dans s et la supprime de s,
- `estVide s` qui teste si il reste des éléments dans s.

L'algorithme de tri proposé est le suivant :

```
let trier liste_a_trier =
  let s = nouvelle()
  and e = ref 0
  and l = ref liste_a_trier
  in
  while not (isEmpty !l) do
    e := head !l;
    l := tail !l;
    inserer s !e;
  done;
  while not (estVide s) do
    e := extraireMax s;
    l := !e :: !l;
  done;
  !l
```

Nous allons étudier le comportement **dans le pire des cas** de cet algorithme en terme de nombre de comparaisons d'éléments suivant les structures de données qu'on peut choisir pour s. On suppose qu'au départ la liste comporte n éléments.

s est un tableau non ordonné.

Q 4.1 Donnez en fonction de i le nombre de comparaisons nécessaires à la i-ème insertion dans le pire des cas.

Q 4.2 Donnez en fonction de i et n le nombre de comparaisons nécessaires à la i-ème extraction dans le pire des cas.

Q 4.3 Donnez le comportement asymptotique du tri. Justifiez.

s est un tableau ordonné.

Q 4.4 Donnez en fonction de i le nombre de comparaisons nécessaires à la i-ème insertion dans le pire des cas.

Q 4.5 Donnez en fonction de i et n le nombre de comparaisons nécessaires à la i-ème extraction dans le pire des cas.

Q 4.6 Donnez le comportement asymptotique du tri. Justifiez.

s est un ABR.

Q 4.7 Donnez en fonction de i le nombre de comparaisons nécessaires à la i-ème insertion dans le pire des cas.

Q 4.8 Donnez en fonction de i et n le nombre de comparaisons nécessaires à la i-ème extraction dans le pire des cas.

Q 4.9 Donnez le comportement asymptotique du tri. Justifiez.

s est un tas (représenté dans un tableau).

Q 4.10 Ecrire la fonction `inserer` qui permet d'ajouter une valeur e à un tas s.

Q 4.11 Donnez en fonction de i le nombre de comparaisons nécessaires à la i-ème insertion dans le pire des cas.

Q 4.12 Donnez en fonction de i et n le nombre de comparaisons nécessaires à la i-ème extraction dans le pire des cas.

Q 4.13 Donnez le comportement asymptotique du tri. Justifiez.

Conclusion

Q 4.14 Quelle structure de données choisiriez-vous ?