

IFT2010 Hiver 2006 — Devoir sans numéro

Miklós Csűrös

9 février 2006

À remettre à la démonstration du 15 février. Ce TP vaut 0 points (mais il sera corrigé).

1 Arbre et tri

Le tri de n éléments prend un temps $\Omega(n \log n)$ dans le pire des cas¹. Montrez que tout algorithme² pour construire un arbre binaire de recherche à partir d'une liste de n éléments doit prendre $\Omega(n \log n)$ temps dans le pire des cas.

2 Un pas en arrière

a. Décrivez l'algorithme $\text{PREDECESSEUR}(x)$ (en pseudo-code ou Java) qui trouve le prédécesseur d'un nœud quelconque x en utilisant les opérations `parent`, `left` et `right` pour se déplacer sur l'arbre. Analysez l'efficacité de l'algorithme.

b. Montrez que quel que soit le nœud de départ dans un arbre binaire de recherche de hauteur h , k appels successifs à PREDECESSEUR dépensent un temps $O(k + h)$. (Les appels successifs sont $y_1 \leftarrow \text{PREDECESSEUR}(x)$; $y_2 \leftarrow \text{PREDECESSEUR}(y_1)$; \dots ; $y_k \leftarrow \text{PREDECESSEUR}(y_{k-1})$.)

¹comparaison de paires des éléments en $O(1)$

²qui utilise la même méthode de comparaison que le tri pour déterminer la relation des nœuds