

PROBLEME 1 : 3

Un arbre est représenté par lien vertical, lien horizontal. Ecrire l'algorithme qui insère un fils de valeur V au point de valeur W s'il existe. Les successeurs d'un point sont rangés par ordre croissant de la valeur.

PROBLEME 2 : Construction d'un tas 9

Un Tas est un arbre binaire parfait, on le stocke donc dans un tableau, comme il est expliqué dans le Polycopié de SDD, tome 2 figure IV.14 (l'arbre est complètement rempli, sauf éventuellement le dernier niveau, où les points sont groupés à gauche). La propriété du Tas est, de plus : la valeur de tout point est \geq à celle de ses deux fils, s'ils existent. Le plus grand élément d'un tas est donc la racine et les sous-arbres d'un nœud contiennent des valeurs inférieures à celle de ce nœud.

Construction d'un Tas à partir de valeurs quelconques contenues dans un tableau T : CONSTRUIRE_TAS(T)

On utilise une procédure RESTAURER_TAS (voir ci-dessous) pour convertir un tableau quelconque $T[1..N]$, en Tas. Comme les éléments de la partie de T allant de $T[N/2 + 1]$ à $T[N]$ sont des feuilles de l'arbre, chacun est au départ un Tas à 1 élément. RESTAURER_TAS est appelée pour chaque nœud de $T[N/2]$ à $T[1]$.

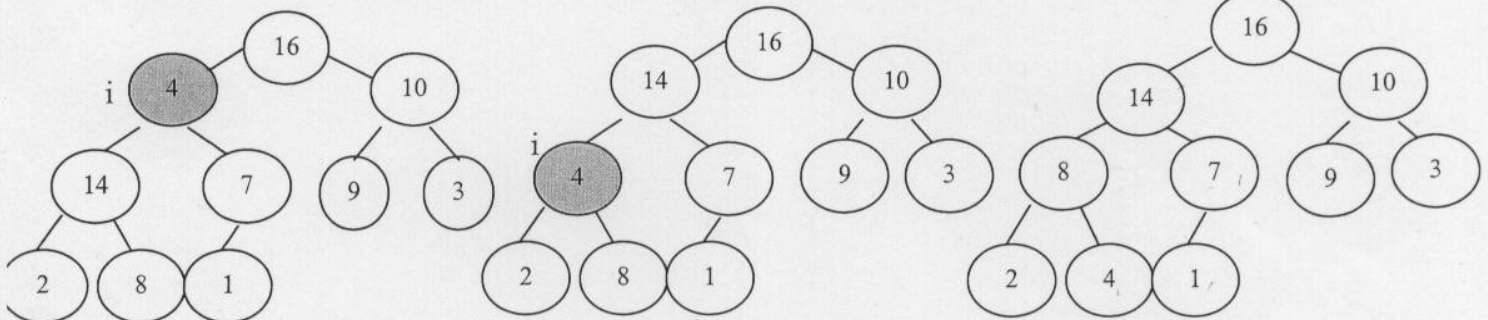
La procédure RESTAURER_TAS (T, i) :

Cette procédure suppose que les sous-arbres gauche et droit d'un nœud d'indice i sont des Tas, mais que le nœud i peut être plus petit que ses fils, violant ainsi la propriété de tas. Le rôle de RESTAURER_TAS est d'échanger le père avec le plus grand de ses 2 fils (si le père n'est pas plus grand que ses fils), de manière à ce que la propriété de tas soit vérifiée en i. Le processus est réitéré jusqu'à ce qu'il n'y ait plus d'échange entre un père et un fils. Exemple :

Etat initial : l'arbre n'est pas un tas

Etat intermédiaire

Etat final : l'arbre est un tas



- 1) Faire un schéma de la structure de données correspondant au tas (état final) de l'exemple précédent.
- 2) Ecrire le principe et l'algorithme de la procédure RESTAURER_TAS (T, i) sous forme itérative.
- 3) Ecrire la procédure CONSTRUIRE_TAS (T)

NB : ne pas utiliser les fonctions m et cm pour ce problème.

PROBLEME 3 4

On gère des blocs d'espace libre de taille variable dans une liste chaînée. Le premier mot de chaque bloc contient sa taille, le dernier mot contient un pointeur vers le bloc suivant.

Donner la procédure qui concatène les blocs d'espace libre adjacents (c'est à dire qui regroupe 2 ou plusieurs blocs lorsqu'ils sont côte à côte) :

- a) dans le cas où la liste d'espace libre est triée par adresses croissantes ;
- b) dans le cas où la liste d'espace libre n'est pas triée.

NB : chaque solution sera commentée, accompagnée d'un schéma clair de la structure (si nécessaire) et d'un lexique des notations utilisées.