Algorithmes et Arbres, $2^{ème}$ Semestre 2011/2012

ÉPREUVE DE RATTRAPAGE, ALGORITHMES ET ARBRES

Durée : 2h, polycopiés et calculatrices autorisés, téléphones interdits.

Exercice 1. (Notations asymptotiques.)
On a la récurrence $T(1) = 1$ et $\forall n > 1, T(n) = n^2 + T(\lfloor n/2 \rfloor)$.
1. Que vaut $T(2^k)$?
2. En déduire l'asymptotique de $T(n)$ quand n tend vers l'infini.

Exercice 2. (Récurrence.)
On cherche à retrouver le pseudo-code du programme **mystère** qui affiche à l'écran les résultats suivant :
mystère(1) affiche 1.
mystère(2) affiche 124.
mystère(3) affiche 124361.
mystère(4) affiche 12436148124.
mystère(5) affiche 124361481245101243 61.

1. Qu'affiche **mystère**(6) ?
2. Écrire un pseudo-code pour le programme **mystère**.
3. Trouver une récurrence pour le nombre d'appel récursif fait par **mystère**(n).

Exercice 3. (Insertion / suppression ABR.) Répondre à chaque question en représentant seulement l'arbre obtenu sans justification.
1. Former un arbre binaire de recherche en insérant successivement, et dans cet ordre, les éléments :

$$13, 8, 4, 10, 15, 5, 14, 17, 2, 16, 11, 20.$$

2. Supprimer dans l'ordre les éléments 8, 15, 13.
3. Faire une rotation gauche en 4.

Exercice 4. (Écrire les entiers uniquement avec des 1.)
On cherche le nombre minimum $T(n)$ de chiffre **1** nécessaire pour "fabriquer" l'entier n en n'utilisant que des $+$, des $\times$, des puissances, et des parenthèses. Exemple : $28 = (1 + 1 + 1)^{(1+1+1)} + 1$ et $T(28) = 7$.
1. Pour n allant de 1 à 12, trouver $T(n)$ et donner une expression utilisant $T(n)$ un.
2. Trouver une récurrence de type programmation dynamique pour $T(n)$.
3. En déduire un $\mathcal{O}$ pour la complexité en nombre de comparaisons pour trouver $T(n)$.
4. Montrer que pour tout $n > 0$, $T(n + 1) \leq T(n) + 1$.

Exercice 5.
Considérons deux séquences croissantes d'entiers $x_1, ..., x_k$ et $y_1, ..., y_k$ et un entier cible C. On cherche un algorithme qui trouve (s'ils existent) x_i et y_j tel que $x_i + y_j = C$.
1. Donner le pseudo-code d'un algorithme en $\mathcal{O}(k)$.