Exercices sur le chapitre 5

<u>Exercice 1</u> On considère un tableau <u>trié</u> tab d'entiers <u>distincts</u> indicé de 1 à n.

a) Trouver un algorithme, dont la complexité temporelle est en $O(\lg(n))$ au pire cas, et qui trouve un indice $1 \le i \le n$, tel que $tab[i] = i$, à condition qu'un tel indice existe.

b) Démontrer la complexité temporelle de votre algorithme avec la méthode générale.

c) Expliquer (informellement) pourquoi votre algorithme est exact.

d) Peut-on trouver un algorithme aussi efficace si les entiers ne sont pas distincts ? Pourquoi ?

<u>Exercice 2</u> On considère l'algorithme d'exponentiation à la russe vu dans le cours (version récursive, page 9 du chapitre 5). Dessiner l'arbre récursif pour $n = 33$ et déterminer le nombre exact de multiplications effectuées.

<u>Exercices numéros 2ef et 4a du devoir 1 de l'automne 2019</u> (Énoncés et solutions sur Moodle)

Solutions

<u>Exercice 1</u>

On considère un tableau <u>trié</u> tab d'entiers <u>distincts</u> indicé de 1 à n.

a)

fonction rechercheT(tab, min, max) **retourne** entier
- entrées :
    - tab : tableau **trié** d'entiers distincts indicé de 1 à n
    - $min \le max$: 2 entiers entre 1 et n
- sortie :
    - s : un entier tel que $tab[s] = s$ ou -1 si un tel indice n'existe pas

début
    si $min = max$ **faire**
        si $tab[min] = min$ **faire**
            $s \leftarrow min$
        sinon faire
            $s \leftarrow -1$
        fin si
    sinon faire
        $m \leftarrow \left\lfloor \frac{min+max}{2} \right\rfloor$
        si $m = tab[m]$ **faire**
            $s \leftarrow m$
        sinon si $m < tab[m]$ **faire**
            $s \leftarrow$ rechercheT(tab, min, m)
        sinon faire
            $s \leftarrow$ rechercheT$(tab, m+1, max)$
        fin si
    fin si
    retourner s
fin rechercheT

b) Au pire cas, on obtient la relation de récurrence $T(n) = T\left(\left\lceil \frac{n}{2} \right\rceil\right) + c$.
$a = 1,\ b = 2, \log_2(1) = 0.$
$f(n) = c = \Theta(n^0)$, donc le cas 2 s'applique, et on obtient ainsi
$T(n) = \Theta\left(n^{\log_b(a)} \lg(n)\right) = \Theta(\lg(n))$.

c) Comme les éléments du tableau sont des entiers triés (en ordre croissant) et distincts, on a forcément $tab[i+1] \ge tab[i] + 1$. Si $tab[i] > i$, il n'est donc plus possible d'avoir $tab[k] = k$ pour $k > i$ (k ne peut plus « rattraper » $tab[k]$). On peut donc se contenter alors de vérifier pour les indices strictement inférieurs à i.

d) Si les entiers ne sont pas distincts, l'information sur un élément du tableau ne peut pas nous aider à éliminer systématiquement la moitié du tableau.

Par exemple pour les trois tableaux $tab_1 = [-1\ 2\ 2\ 6\ 9\ ]$, $tab_2 = [-1\ 0\ 2\ 4\ 7\ ]$ et $tab_3 = [-1\ 0\ 2\ 6\ 9\ ]$, on a $tab[3] < 3$. Pourtant, on obtient alors successivement $s_1 = 2 < 3$, $s_2 = 4 > 3$ et $s_3 = -1$.

<u>Exercice 2</u>

8 multiplications.