TD 5 - Tri et complexité

Objectif : Appliquer, optimiser, analyser et concevoir un algorithme de tri.

Exercice 1 : On rappelle ci-dessous l'algorithme du drapeau à 3 couleurs vu en cours :

Algorithme 1 : Algorithme du drapeau à 3 couleurs

```
début
    /* ENTRÉES : Un vecteur V de taille n */
    /* SORTIE : Le vecteur V trié */
    i ← 1 ; j ← 1 ; r ← n
    tant que i ≤ r faire
        si V(i) = J alors i ← i + 1
        sinon
            si V(i) = B alors
                Echange(V, j, i)
                j ← j + 1
                i ← i + 1
            sinon
                tant que V(r) = R et r > i faire r ← r − 1
                Echange(V, r, i)
                r ← r − 1
    retourner V
fin
```

a) Appliquer cet algorithme à l'instance $V = [J,R,R,B,J,J,J,R,B]$

b) Calculer la complexité en nombre de comparaisons et d'échanges dans le pire et le meilleur cas, en fonction de n_B, n_J et n_R désignant respectivement le nombre d'éléments bleus, jaunes et rouges. On a $n_B + n_J + n_R = n$.

Exercice 2 : Optimisation du tri à bulles

On a vu que l'algorithme du tri à bulles pouvait être optimisé en s'arrêtant dès qu'il n'y a aucun échange lors d'une itération. D'autres améliorations sont possibles :

a) Appliquer l'algorithme sur le vecteur $[12,9,3,1,18,43]$. A la deuxième itération, était-il nécessaire de parcourir le vecteur de $j = 1$ à $n-2$? Proposer un algorithme modifié pour ne pas aller plus loin que nécessaire à chaque itération.

b) Appliquer l'algorithme sur les vecteurs $[2,3,4,5,6,1]$ et $[6,1,2,3,4,5]$. Vous remarquez une nette différence de complexité alors que dans les deux cas un seul élément est mal placé. Nous allons construire un nouvel algorithme :
   — Pour le premier vecteur, que se passe-t-il si à chaque itération on alterne le sens du tri à bulles ?
   — Tester cette procédure sur le vecteur $[12,9,3,1,18,43,2]$, en notant bien les indices de dernière permutation dans chaque sens, respectivement i_{GD} et i_{DG}. A chaque itération, quel doit être l'intervalle de variation de j ? Quelle est la condition d'arrêt sur i_{GD} et i_{DG} ?
   — Ecrire l'algorithme.

Exercice 3 : tri par insertion

Écrivez l'algorithme de tri par insertion pour une liste. Les opérations disponibles sont *tete*, *succ*, *pred*, *esttete*, *estqueue*, *insere_avant*, *insere_apres*. Ces deux dernières opérations prennent deux arguments : l'élément à insérer et respectivement celui avant lequel et celui après lequel on insère.

Exercice 4 : On rappelle que pour deux entiers a et b, $a \mod b$ désigne le reste de la division entière de a par b. Soit l'algorithme suivant :

Algorithme 2 : Algorithme X

```
début
    /* ENTRÉES : un entier x et un vecteur V de n entiers trié en ordre décroissant*/
    /* SORTIE : le vecteur V modifié */
    i ← 1
    m ← n
    tant que i ≤ m et V(i) ≥ x faire
        si V(i) mod x = 0 alors
            pour j = i → m − 1 faire
                V(j) ← V(j + 1)
            m ← m − 1
        sinon
            i ← i + 1
    retourner V
fin
```

a) Déroulez cet algorithme sur un exemple (cet exemple doit illustrer clairement le fonctionnement et le rôle de l'algorithme). Quel est son rôle ?

b) La condition $V(i) \geq x$ est-elle nécessaire ? Si non, quel est son intérêt ?

c) Quelle(s) est (sont) le(s) opération(s) significative(s) à utiliser pour évaluer la complexité de cet algorithme ?

d) Calculez la complexité de cet algorithme en pire cas et en meilleur cas.