

TD 4 - Tri et Complexité

Objectif : Savoir dérouler un algorithme et calculer sa complexité.

Exercice 1 - Appliquer l'algorithme du tri par comptage sur le mot "ZEUGME".

Exercice 2 - On considère l'algorithme de recherche dichotomique de la première occurrence, rappelé ci-dessous.

1. Quelles sont les opérations significatives ? Exprimer la complexité C en fonction du nombre d'itérations p .
2. Soit n_k la longueur de l'intervalle $[[inf_k, sup_k]]$ à la $k^{\text{ème}}$ itération. Calculer n_{k+1} en fonction de n_k , selon le test " $x \leq V(med_k)$ " et selon la parité de n_k .
3. Sachant que la suite des n_k est strictement décroissante et que la dernière valeur $n_p = 1$ (quand inf et sup deviennent égaux) :
 - Comment doit décroître la suite des n_k pour avoir le nombre maximal d'itérations ? Calculer dans ce cas n_{p-k} en fonction de k . Sachant que $n_0 = n$, en déduire p en fonction de n , puis la complexité dans le pire cas C_{max} .
 - Comment doit décroître la suite des n_k pour avoir le nombre minimal d'itérations ? Calculer n_{p-k} en fonction de n , en déduire p en fonction de n , puis la complexité dans le meilleur cas C_{min} .
4. En déduire l'ordre de la complexité moyenne.

Algorithme 1 : Recherche dichotomique d'une 1ère occurrence dans un vecteur trié

début

/* ENTRÉES : Un vecteur V de taille n , un élément x */

/* SORTIE : i si x apparaît au rang i de V , 0 si $x \notin V$ */

$inf \leftarrow 1, sup \leftarrow n, i \leftarrow 0$

tant que $inf < sup$ **faire**

$med \leftarrow (inf + sup) \text{ div } 2$

si $x \leq V(med)$ **alors** $sup \leftarrow med$

sinon $inf \leftarrow med + 1$

si $V(inf) = x$ **alors**

retourner inf

sinon

retourner 0

fin

Exercice 3 - Pour chacun des algorithmes suivants, déterminez son rôle, les opérations significatives à considérer et la complexité dans le pire cas et dans le meilleur cas. Comment pourrait-on optimiser l'algorithme 3 ?

Algorithme 2 :

```
début
  /* ENTRÉES : Une pile  $P$ , un élément  $x$  */
  /* SORTIE : A DETERMINER */
   $trouve \leftarrow Faux$ 
  while not( $est\_vide(P)$ ) et not  $trouve$  do
    si  $x = sommet(P)$  alors  $trouve \leftarrow Vrai$ 
    else  $depile(P)$ 
  retourner  $trouve$ 
fin
```

Algorithme 3 :

```
début
  /* ENTRÉES : Deux piles  $P$  et  $Q$  */
  /* SORTIE : A DETERMINER */
   $T := pilevide$ 
   $U := pilevide$ 
  while not( $est\_vide(P)$ ) do
     $trouve \leftarrow Faux$ 
     $x \leftarrow sommet(P)$ 
     $depile(P)$ 
    while not( $est\_vide(Q)$ ) do
       $e \leftarrow sommet(Q)$ 
       $depile(Q)$ 
       $empile(e, U)$ 
    si  $x = e$  alors  $trouve \leftarrow Vrai$ 
    while not( $est\_vide(U)$ ) do
       $e \leftarrow sommet(U)$ 
       $depile(U)$ 
       $empile(e, Q)$ 
    si  $trouve$  alors  $empile(x, T)$ 
  retourner  $T$ 
fin
```
