

Preuve et Analyse des Algorithmes

Module I41, Licence Informatique, USTV.

Novembre 2010

↯ Le sujet est composé 7 exercices à traiter en moins de 0x5A minutes. Aucun document autorisé. Vous êtes invités à remettre une copie claire, concise, sans rature ni surcharge. Il est par ailleurs inutile de recopier l'énoncé. . . La note finale tiendra compte de la présentation générale de la copie.

1 Sommes

1. Donner la valeur de

$$1 + 2 + \dots + 1000$$

2. Vrai ou faux ? Justifier.

$$1 + 3 + 3^2 + \dots + 3^{n-1} \sim 3^n$$

2 Notation asymptotique

Soient f et g deux applications de \mathbf{N} dans \mathbf{N} .

1. Donner la définition de $f = O(g)$.
2. Montrer que si $f = O(g)$ alors $g = \Omega(f)$.

3 Temps de calcul

Une implantation d'un algorithme $A(n:\text{entier})$ dont le temps de calcul est quadratique en n se termine en 3 secondes pour l'instance $n=1024$. Estimer le temps de d'exécution de $A(2048)$.

4 Dichotomie

Ecrire une fonction (en langage C) : `double solution(a, b, e : double)`, pour résoudre à e -près l'équation $f(x) = 0$ sur l'intervalle $[a, b]$ sachant que f est continue sur $[a, b]$, $f(a) > 0$ et $f(b) < 0$.

5 Algorithme du cours

Décrire le principe “diviser pour régner”.

6 Analyse

1. Que l’algorithme ci-dessous ?
2. Soit $C(n)$ le nombre de comparaisons entre objets. Déterminer $C(n)$ en fonction de n .

```
1
2 Algorithme tri( t : tableau)
3 variable i , j, n: indice
4         tmp : objet;
5 debut
6   n := taille( t );
7   tantque ( i < n )
8     j := i+1;
9     tantque ( j < n )
10      si ( t[i] > t[j] ) alors
11        tmp := t[i];
12        t[i] := t[j];
13        t[j] := tmp;
14      fsi
15      j := j + 1;
16    ftq
17  i := i + 1
18 ftq
19 fin
```

7 Invention

Soit t un tableau de n objets. Un objet v est dit majoritaire dans t quand il apparaît plus de $\frac{n}{2}$ fois dans t .

1. Un tableau possède au plus un objet majoritaire. Pourquoi ?
2. Ecrire un algorithme naïf pour déterminer si t possède un objet majoritaire.
3. Estimer le temps de calcul de l’algorithme.
4. Ecrire un algorithme plus efficace.