

TD d'Algo n° 1

EPITA ING1 2013

1 Compte-tours

1.1 Bouclettes

Combien de lignes sont affichées par les boucles suivantes ? Donnez votre réponse en fonction de N et n'hésitez pas à distinguer les cas où N est pair de ceux où il est impair.

```
for (int i = 0; i < N; ++i)
    for (int j = 1; j <= N; ++j)
        puts("Boucle 1");

for (int i = 0; i < N; ++i)
    for (int j = N/2; j > 0; --j)
        puts("Boucle 2");

for (int i = 0; i < N; ++i)
    for (int j = i; j >= 0; j--)
        puts("Boucle 3");

for (int i = 0; i < N; ++i)
    for (int j = i; j < N - 2; j++)
        puts("Boucle 4");

for (int i = 0; i < N; ++i)
    for (int j = i; j >= 0; j -= 2)
        puts("Boucle 5");

for (int i = 0; i < N; ++i)
    for (int j = 1; j < N; j *= 2)
        puts("Boucle 6");
```

1.2 Maison

Combien de caractères '#' sont affichés par la fonction suivante ? Donnez votre réponse en fonction de n

```
1 void house(unsigned int n)
2 {
3     for (unsigned int y = 0; y < n; ++y)
4         {
5             for (unsigned int x = 1; x < n - y; ++x)
6                 putchar(' ');
7             for (unsigned int x = n - y; x <= n + y; ++x)
8                 putchar('#');
9             putchar('\n');
10        }
11    for (unsigned int y = 1; y < n; ++y)
12        {
13            for (unsigned int x = 1; x < 2*n; ++x)
14                putchar('#');
15            putchar('\n');
16        }
17 }
```

2 Notions d'arbres (binaires)

On considère des arbres binaires non vides dont les nœuds ont soit deux fils (nœuds internes) soit aucun fils (feuilles). On note h la hauteur de l'arbre ($h = 0$ pour un arbre constitué d'un seul nœud), n_i le nombre de nœuds internes, f le nombre de feuilles, et $n = n_i + f$ le nombre de nœuds total.

Exprimez n_i en fonction de f .

Encadrez (pensez qu'un arbre peut être très déséquilibré) :

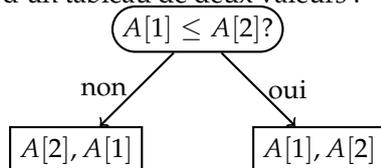
- f en fonction de h ,
- n_i en fonction de h ,
- n en fonction de h .

Déduisez-en des minorants (les hauteurs étant forcément des entiers, utilisez les parties entières à bon escient) pour :

- h en fonction de f ,
- h en fonction de n_i ,
- h en fonction de n .

3 Arbres de décision

Il s'agit de construire un arbre qui représente un algorithme de tri pour un tableau de taille n connue. Chaque nœud interne de l'arbre correspond à une question de la forme " $A[i] \leq A[j]$?" (pour des i et j donnés). Le fils gauche d'un tel nœud est suivi si la réponse est négative, sinon, on emprunte le fils droit. Chaque feuille correspond à une réponse possible de l'algorithme, c'est-à-dire l'ordre dans lequel il faudrait mettre le tableau pour avoir ces valeurs dans l'ordre (plusieurs feuilles peuvent donner la même réponse). Par exemple voici un tri d'un tableau de deux valeurs :



1. Dessinez un tel arbre pour trier un tableau de 3 valeurs.
2. Pour un tableau de taille n , combien cet arbre devrait-il avoir de feuilles au minimum ?
3. Déduisez-en la hauteur minimale d'un tel arbre pour trier n valeurs.
4. Qu'en déduire sur la complexité des tris comparatifs ?

4 Calculs simples pour ne pas perdre la main

1. Calculez la somme des 1000 premiers entiers naturels impairs.
2. Calculez $\log_{10}(42)$ avec deux chiffres après la virgule, en vous aidant éventuellement d'une calculatrice 4 opérations (touche "log" interdite!).
3. Calculez $\log_2(42)$ de même.

Le TP de cet après-midi et des jours suivants

Ça se passe au *Mid Lab*, au *Lab SR*, et en *SM 14*. Après vous être installé, connectez-vous sur :

<http://algo.lrde.epita.fr/>

et cliquez sur le titre du TP1 pour lire le sujet.