Travaux Dirigés n°3

Cours d'Informatique de Deuxième Année

—Deug SSM M3-MI—

Complexité d'algorithme

Nous montrons des techniques de résolution d'équation de complexité.

---

1 Récursivité

Etant donné un terme t et une fonction f, une fonction primitive récursive F est une fonction de la forme :

$$F(n) = \begin{cases} t & \text{si } n = 0, \\ f(n, F(n-1)) & \text{sinon et } f \text{ est elle-même primitive récursive.} \end{cases}$$

Par exemple, la fonction factorielle `fact` est définie à partir de la fonction de multiplication `mult` qui est elle-même primitive récursive :

$$\texttt{fact}(n) = \begin{cases} 1 & \text{si } n = 0, \\ \texttt{mult}(n, \texttt{fact}(n-1)) & \text{sinon.} \end{cases}$$

▸ **Exercice 1. (La fonction d'Ackerman)** *La fonction d'Ackerman* `Ack` *ne fait pas partie des fonctions primitives récursives (pourquoi ?).*

$$\texttt{Ack}(n, m) = \begin{cases} 1 & \textit{si } m = 0 \textit{ et } n \geq 1, \\ 2m & \textit{si } n = 0, \\ \texttt{Ack}(n-1, \texttt{Ack}(n, m-1)) & \textit{sinon.} \end{cases}$$

Ecrire la fonction `Ack`*. Effectuer "à la main" le calcul de* $\texttt{Ack}(2,2)$, $\texttt{Ack}(2,3)$*. Que pensez-vous du calcul de* $\texttt{Ack}(3,3)$ *?*

2 Complexité

▸ **Exercice 2. (Estimation d'une complexité en temps simple)** *Soit le programme C suivant :*

```
#include <stdio.h>
...
int SousSommeMax(int t[], int nb)
{ unsigned i,j,k;
  int S, max;
  for (i=0; i<nb ; i++)
    for (j=i; j<nb; j++)
    { S=0;
      for (k=i; k<=j; k++) S+=t[k];
      max=S>max?S:max;
    }
  return max;
}

void main (void)
{ int T[1000], nb;
  LireTab(T,&nb);
  printf("sous-somme max: %d\n", SousSommeMax(T,nb));
}
```

On suppose la fonction `LireTab` *écrite (de toute manière, la question n'est pas là...). Donnez la complexité en temps de la fonction* `SousSommeMax`*. Déduisez-en une approximation du temps d'exécution du programme ci-dessus.*

▸ **Exercice 3. (Complexité de la fonction** `Dichotomie`**)** *Calculez la complexité en temps de la fonction* `Dichotomie` *du 1er exercice. Comparez les performances de la fonction* `Dichotomie` *à celles d'une simple fonction de recherche qui parcourt tout le tableau.*

▸ **Exercice 4. (Résolution d'équation)** *Soit l'équation :*

$$T(n) = \begin{cases} 1, & si\ n = 1, \\ 2 \times T(\lfloor \frac{n}{2} \rfloor) + n, & sinon. \end{cases}$$

Donnez une approximation de la complexité de la fonction T.