

TP 1

Calcul de la sous-somme maximale d'un tableau

1 Introduction

Etant donné un tableau T de n entiers, on cherche à savoir quels sont les deux indices i et j tels que la somme des éléments de i à j dans T soit maximale. Les éléments de T peuvent évidemment être négatifs, autrement la solution est trivialement le début et la fin du tableau.

Exemple : $T = [-10, -8, 18, 9, 21, 9, 22, 14, -1, -12]$

La solution est $i = 3$, $j = 7$, et vaut 93.

2 Première méthode : naïve

Le premier algorithme consiste à parcourir toutes les sous-séquences, et à garder celle dont la somme est maximale.

Question 1 : Ecrire l'algorithme naïf.

Question 2 : Evaluer sa complexité théorique.

3 Seconde méthode : diviser pour régner

Ici, on considère que l'on est forcément dans l'un des cas suivant :

- la sous-séquence maximale est incluse dans la première moitié du tableau
- la sous-séquence maximale est incluse dans la seconde moitié du tableau
- la sous-séquence maximale est à cheval sur les deux parties du tableau

Pour les deux premiers cas, il suffit de rappeler la méthode (appel récursif) sur une des deux moitiés du tableau. Concernant le troisième cas, il suffit de calculer (itérativement) :

- la sous-séquence maximale dont la limite droite est le milieu du tableau
- la sous-séquence maximale dont la limite gauche est le milieu du tableau + 1

On fusionne alors les deux sous-séquence obtenues, et on décide quelle est la meilleure sous-séquence parmi les trois que l'on a calculé.

Question 3 : Ecrire l'algorithme ainsi défini.

Question 4 : Evaluer sa complexité théorique.

4 Implémentation

Implémenter les deux algorithmes ainsi définis en langage C/C++. Le programme prendra en entrée un entier qui correspondra à la taille du tableau. On générera ensuite un tableau de cette taille rempli à l'aide d'entiers aléatoires appartenant à l'intervalle $[-25, 25]$. On exécutera ensuite les deux algorithmes sur ce tableau, et on affichera le temps d'exécution de chacun.

Exemple d'utilisation du programme et affichage généré :

```
./maxSousTableau 100000
```

```
i=89702, j=90938 s=1086
```

```
i=89702, j=90938 s=1086
```

```
Premier algorithme: 12.500 s
```

```
Deuxieme algorithme: 0.022 s
```