

# Rapport de TP 4MMAOD : Génération d'ABR optimal

MILAN Guillaume (SLE) KOUVTANOVITCH Geoffrey (SLE)

April 14, 2017

## Préambule

- Compléter ce patron de rapport en supprimant toutes les phrases en italique : elles ne doivent pas apparaître dans le rapport pdf.
- Il sera attribué 1 point pour la qualité globale du rapport : présentation, concision et clarté de l'argumentation.

## 1 Principe de notre programme (1 point)

L'objectif du programme est d'obtenir l'arbre de parcour optimal sur l'ensemble des éléments. Pour cela nous avons donc cherché à obtenir la racine optimal de notre arbre. Mais par l'expression évoqué dans le rapport précédent liant un arbre optimal a ses sous arbres droite et gauche. Ainsi pour trouver la racine de optimal l'arbre contenant les éléments  $i$  à  $j$  il suffit de parcourir tout les éléments  $k$  entre  $i$  et  $j$  et en choisissant  $k$  comme racine de l'arbre. Cet arbre a donc pour profondeur moyenne d'après la formule du rapport précédent:  $P_k(i, j) = P(i, k - 1) * P_{tot}(i, k - 1) + P_{tot}(k, k) + P(k + 1, j) * P_{tot}(k + 1, j)P_{tot}(i, j)$  on en déduit que pour cela il faut connaître les arbre optimaux de  $(i, k - 1)$  et  $(k + 1, j)$  pour tout  $k$ . Ceci implique de le calculer avant. Nous avons donc fait le choix d'organiser notre programme avec une structure de donné contenant la profondeur moyenne d'un arbre optimal, sa racine et son poids total. Cette structure est alors utilisé pour fabriquer une matrice (cf Figure:1) diagonal supérieur qui stocke les information des arbre optimaux. Ici  $M(i, j)$  contient les données de l'arbre optimal entre  $i$  et  $j$ .

*Pour des raison de clarté et d'homogénéité au programme, la matrice a ses indices qui commence à 0*

Une fois cette matrice calculé il suffit de construire l'arbre optimal en partant de la racine  $M(0, n - 1)$  avec  $n$  la taille de la matrice en respectant le principe de construction d'arbre optimal. C'est à dire que tout sous arbre est optimal. (cf Figure:2 et Figure:3).

## 2 Analyse du coût théorique (2 points)

*Donner ici l'analyse du coût théorique de votre programme en fonction du nombre  $n$  d'éléments dans le dictionnaire. Pour chaque coût, donner la formule qui le caractérise (en justifiant brièvement pourquoi cette formule correspond à votre programme), puis l'ordre du coût en fonction de  $n$  en notation  $\Theta$  de préférence, sinon  $O$ .*

### 2.1 Nombre d'opérations en pire cas :

**Justification :** *La justification peut être par exemple de la forme:*

*"Le programme itératif contient les boucles  $k_1 = \dots, k_2 = \dots$  etc correspondant à la somme*

$$T(n_1, n_2, c_1, c_2) = \sum_{k_1=\dots}^{\dots} \dots \sum \dots + \sum_{i=\dots}^{\dots} \dots$$

*somme que nous avons calculée (ou majorée) par la technique de ... "*

*ou encore :*

*"les appels récursifs du programme permettent de modéliser son coût par le système d'équations aux récurrences*

$$T(k_1, k_2) = \dots \text{ avec les conditions initiales } \dots$$

*Le coût indiqué est obtenu en résolvant ce système par la méthode de .... "*

### 2.2 Place mémoire requise :

**Justification :**

## 2.3 Nombre de défauts de cache sur le modèle CO :

**Justification :**

## 3 Compte rendu d'expérimentation (2 points)

### 3.1 Conditions expérimentales

*Décrire les conditions permettant la reproductibilité des mesures: on demande la description de la machine et la méthode utilisée pour mesurer le temps.*

#### 3.1.1 Description synthétique de la machine :

*indiquer ici le processeur et sa fréquence, la mémoire, le système d'exploitation. Préciser aussi si la machine était monopolisée pour un test, ou notamment si d'autres processus ou utilisateurs étaient en cours d'exécution.*

#### 3.1.2 Méthode utilisée pour les mesures de temps :

*préciser ici comment les mesures de temps ont été effectuées (fonction appelée) et l'unité de temps; en particulier, préciser comment les 5 exécutions pour chaque test ont été faites (par exemple si le même test est fait 5 fois de suite, ou si les tests sont alternés entre les mesures, ou exécutés en concurrence etc).*

### 3.2 Mesures expérimentales

*Compléter le tableau suivant par les temps d'exécution mesurés pour chacun des 6 benchmarks imposés (temps minimum, maximum et moyen sur 5 exécutions)*

	coût du patch	temps min	temps max	temps moyen
benchmark1				
benchmark2				
benchmark3				
benchmark4				
benchmark5				
benchmark6				

Figure 1: Mesures des temps minimum, maximum et moyen de 5 exécutions pour les 6 benchmarks.

### 3.3 Analyse des résultats expérimentaux

*Donner une réponse justifiée à la question: les temps mesurés correspondent ils à votre analyse théorique (nombre d'opérations et défauts de cache) ?*