Rapport de TP 4MMAOD : Génération d'ABR optimal

Maxime Deloche
Ludovic Carré
Equipe Teide 4

11 octobre 2016

1 Equation de Bellman

1.1 Question 1

1.1.1 Montrons que tout sous-arbre d'un ABR optimal est un ABR optimal, par l'absurde.

Soit A un ABR optimal. On suppose qu'un sous-arbre S de A n'est pas optimal. Donc le nombre de comparaisons x pour atteindre un élément e de S n'est pas optimal dans S (notons cet optimal x_{opt}). Donc le nombre de comparaisons pour atteindre e dans A est $x+1$. Or, on pourrait atteindre e en $x_{opt}+1$, donc A n'est pas optimal, ce qui est absurde.

Donc tout sous-arbre d'un ABR optimal est un ABR optimal.

1.1.2 Equation de Bellman

Deux raisonnements différents nous ont conduit à deux équations de Bellman différentes mais équivalentes. Nous les avons implémentées et on obtient des résultats identiques et corrects.

- **Version 1**
  Dans un arbre A de racine e_i, on a :
  - une probabilité `proba_que_element_soit_e_i` de faire 1 comparaison (si l'élément cherché est e_i).
  - une probabilité `proba_que_element_soit_dans_le_sous_arbre_gauche` de faire le nombre de comparaisons optimal du sous-arbre gauche, plus 1.
  - une probabilité `proba_que_element_soit_dans_le_sous_arbre_droit` de faire le nombre de comparaisons optimal du sous-arbre droit, plus 1.

  Soit $c(E)$ le nombre de comparaisons moyen optimal de l'arbre formé par les éléments de E. L'équation de Bellman est :

$$C(E) = \min_{e_i \in E}\left(\frac{p_i}{\sum_{e_l \in E} p_l} + \frac{C(\{e_j < e_i\})+1}{\sum_{e_l \in E} p_l}\sum_{e_j < e_i} p_j + \frac{C(\{e_k > e_i\})+1}{\sum_{e_l \in E} p_l}\sum_{e_k > e_i} p_k\right)$$

  C'est-à-dire :

$$C(E) = \min_{e_i \in E}\left(1 + \frac{C(\{e_j < e_i\})}{\sum_{e_l \in E} p_l}\sum_{e_j < e_i} p_j + \frac{C(\{e_k > e_i\})}{\sum_{e_l \in E} p_l}\sum_{e_k > e_i} p_k\right)$$

  Conditions initiales :
  — $C(E) = 1$ si E ne contient qu'un élément (on ne fait alors qu'une comparaison).
  — $C(E) = 0$ si E est vide.

- **Version 2**
  En transformant la formule donnée de calcul du nombre de comparaison moyen optimal en version récursive : Soit P la profondeur, initialisée à 1. L'équation de Bellman est :

$$C(E,P) = \min_{e_i \in E}(p_i.P + C(\{e_j < e_i\}, P+1) + C(\{e_k > e_i\}, P+1))$$

  Conditions initiales :
  — $C(E) = p_i.P$ si $E = e_i$.
  — $C(E) = 0$ si E est vide.

1.2 Question 2

- **Coût de la version 1**
  - **En mémoire** : on stocke dans un tableau de taille n par n le nombre moyen optimal de comparaisons, ainsi que le noeud choisi en racine pour cet arbre. Par exemple, la position (i, j) de ce tableau contiendra le nombre de comparaisons et la racine de l'abre formé par les éléments e_i à e_j (en supposant les e_i ordonnés). On a donc un coût en mémoire de $\theta(n^2)$.

  - **En temps** : Pour chaque appel, on a de l'ordre de k^2 opérations à effectuer, avec k le nombre d'éléments de E. On remarque qu'on a $(n-k)$ appels avec k éléments, donc le coût est $\sum_{k=0}^{n}(n-k)k^2 = \frac{1}{12}n(n+1)(n^2+3n+2)$. On a donc un coût en $\theta(n^4)$.

- **Coût de la version 2**
  - **En mémoire** : On a le même coût en mémoire qu'à la version 1 puisque la valeur optimale ne dépend pas de la profondeur. On est donc en $\theta(n^2)$.
  - **En temps** : Pour un seul appel, on a un coût en $\theta(1)$ et puisque l'on a n^2 appels, on a un coût total de l'ordre de $\theta(n^2)$.

Conclusion : On retient donc l'équation de la version 2 puisque elle propose une complexité plus avantageuse.