1 Exercice 1 : ABR et aléas

On considère un ensemble de clés $\mathcal{K} \subseteq \mathbb{N}$ fini de taille m. On note $\mathcal{A}_\mathcal{K}$ l'ensemble des arbres binaires dont les noeuds sont étiquetés par des éléments *distincts* de $\mathcal{K}$. Pour $A \in \mathcal{A}_\mathcal{K}$, on notera $\kappa(A)$ l'ensemble des étiquettes des nœuds de A. Par convention, on dira que la racine de A est de profondeur 0.

2.1 Rappeler la définition inductive d'un *arbre binaire de recherche* (ABR).

2.2 Rappeler la complexité dans le pire des cas de la recherche dans un ABR ayant n nœuds.

Soit $A \in \mathcal{A}_\mathcal{K}$ tel que $\kappa(A) = \mathcal{K}$. Soit $\mathbb{P}$ une loi de probabilité quelconque, sur les clés de $\mathcal{K}$. On définit H_A la variable aléatoire qui, à une clé de $\mathcal{K}$, associe sa profondeur dans A. Ainsi $H_A(k)$ est la profondeur de k pour une clé k.

2.3 Soit un ABR fixé $A \in \mathcal{A}_\mathcal{K}$. Rappeler l'expression de la complexité en moyenne de la recherche d'une clé dans A.

2.4 On suppose ici que $\mathbb{P}$ est la loi uniforme sur les clés de $\mathcal{K}$. Quelles sont les particularités des ABR $A \in \mathcal{A}_\mathcal{K}$ pour lesquels $\mathbb{E}(H_A)$ est minimale. Montrer que $\mathbb{E}(H_A) = O(\log(m))$.

On dit que $A \in \mathcal{A}_\mathcal{K}$ est optimal pour $\mathcal{K}$ et $\mathbb{P}$ si et seulement si $\kappa(A) = \mathcal{K}$ et il minimise $\mathbb{E}(H_A)$.

On cherche un algorithme calculant un ABR optimal étant donné un ensemble de clés fini et une loi de probabilité sur ces clés.

2.5 De quel type de problème algorithmique s'agit-il?

2.6 Pour cette question, on suppose $\mathcal{K} = \{1,2,3\}$, $\mathbb{P}(1) = \frac{2}{3}$, $\mathbb{P}(2) = \mathbb{P}(3) = \frac{1}{6}$. Dessiner un ABR optimal pour $\mathcal{K}$ et $\mathbb{P}$

2.7 Soit $A \in \mathcal{A}_\mathcal{K}$ un ABR optimal non vide. A possède donc un sous-arbre gauche A_g. On se place dans le cas où A_g est non vide. On note $S_g = \sum_{k \in \kappa(A_g)} \mathbb{P}(k)$ et $\mathbb{P}_g = \frac{1}{S_g}\mathbb{P}$. Montrer que A_g est optimal pour $\kappa(A_g)$ et $\mathbb{P}_g$. Montrer le résultat analogue pour le sous-arbre droit.

2.8 Proposer un algorithme de programmation dynamique pour trouver un ABR optimal.

2 Exercice 2 : simuler un tirage aléatoire

1. On considère un algorithme `biaise` qui renvoie 1 avec une probabilité p inconnue et 0 sinon. Considérons l'algorithme de Las Vegas suivant : on utilise deux fois consécutives la fonction `biaise`, si on obtient 10 alors on renvoie 0, si on obtient 01, on renvoie 1 et sinon on recommence jusqu'à obtenir l'un ou l'autre des motifs cherchés.
   (a) Justifier que la probabilité que cet algorithme ne s'arrête pas est nulle.
   (b) Quelle est la probabilité de renvoyer 0 ? 1 ?
2. On suppose disposer d'une fonction `pile_ou_face` qui renvoie 0 et 1 avec une probabilité identique qui vaut 1/2.
   Soit p un réel de $]0,1[$. En exploitant la décomposition de p en base 2, déterminer un algorithme qui renvoie 0 avec probabilité p et 1 avec probabilité $1-p$ en utilisant la fonction `pile_ou_face`.
3. On suppose qu'on dispose d'une fonction `foo()` qui renvoie 0 avec probabilité $\frac{1}{2}$ et 1 aussi avec probabilité $\frac{1}{2}$.
   (a) Décrire une manière de tirer un entier uniformément entre 0 et 3 en utilisant cette fonction.
   (b) Soit $n \in \mathbb{N}^*$. Généraliser le résultat précédent pour trouver un nombre entre 0 et $2^n - 1$.
   (c) Donner une méthode pour générer un entier uniformément entre 0 et 5 en utilisant toujours uniquement la fonction `foo`.
   (d) Supposons maintenant qu'on dispose d'une fonction `foo_bis()` qui renvoie uniformément un entier entre 0 et 4. Décrire une méthode pour obtenir uniformément un entier entre 0 et 6 en utilisant uniquement cette dernière fonction.

Corrigé exo 1

1 arbre binaire avec noeuds étiquettés avec des valeurs t.q pour tt noeud sa valeur est plus gde que ttes celles contenues ds le sous arbre gauche et plus petite que ttes celles contenues ds le ss arbre dt.

2 Au pire cas, une branche peut tout à fait être de longueur n et le coût de la recherche étant la hauteur de l'arbre celui ci peut donc valoir n.

3 $\sum_{k \in K} H_A(k) \times \mathbb{P}(k) = E(H_A)$

4 Si $\mathbb{P}$ est uniforme alors $E(H_A)$ est minimale pour un arbre parfait (complet sauf éventuellement le dernier niveau).

On va m.q, ds ce cas, $E(H_A) = O(\log m)$

Il y a une racine

2 noeuds à prof 1 ; et au plus 2^h à hauteur h avec $h = \lfloor \log_2 m \rfloor$

4 ——— 2

2^k ——— k

2^{h-1} ——— h-1

On obtient donc :

$$\frac{1}{m}\left(\sum_{k=0}^{h-1} k 2^k + h \text{ nb de feuilles}\right)$$

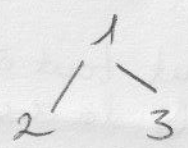
$$\leq \frac{1}{m}\left(\frac{-2^h + 1}{1-2} \times h + 2^h h\right) \leq \frac{1}{m}\left(2^{h+1} - 1\right) \times h$$

est $O\left(\frac{2^h}{m} \times h\right)$ avec

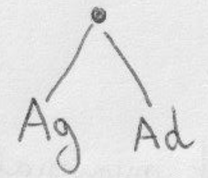
$O\left(\frac{2^h}{m}\right) = O(1)$ donc $O(h) = O(\lfloor \log(m) \rfloor)$

(5) pb d'optimisat°.

(6)



(7)



Supposons que A_g n'est pas optimal pour $K(A_g)$ et $\mathbb{P}_g$ par l'absurde et donc il existe A_g' t.q $E(H_{A_g'}) < E(H_{A_g})$

Estimons $E(H_A)$ en fct de $E(H_{A_g})$ et $E(H_{A_d})$:

$$E(H_A) = \sum_{k \in K} \mathbb{P}(k) \times H_A(k)$$

$$= \sum_{k \in K(A_g)} \mathbb{P}(k) \times (H_{A_g}(k) + 1) +$$

$$\sum_{k \in K(A_d)} \mathbb{P}(k) \times (H_{A_d}(k) + 1)$$

$$= S_g\left(E(H_{A_g}) + 1\right) + S_d\left(E(H_{A_d}) + 1\right)$$

et donc remplace A_g par A_g' donnerait un ABR meilleur ce qui contredirait l'optimalité de A_g.

(8) On pose E_{ij} l'espérance d'un arbre contenant les clés $k_i \dots k_j$ avec $k_1 \leq \dots \leq k_m$

on a

$E_{ii} = 0$ (réduit à une racine)

et $E_{ij} = 0$ avec $j \leq i$

$$E_{ij} = \min_{r \in [\![i,j]\!]} \left(S_{i,r-1}(E_{i,r-1} + 1), (E_{r+1,j} + 1) S_{r+1,j} \right)$$

avec $S_{i,j} = \sum_{k_i}^{k_j} \mathbb{P}(k)$

<u>On calcule $E_{1,m}$</u>

(9) $O(m^3)$

<u>Exercice 2</u>

(1) (a) A chaque tour, la proba de recommencer vaut $p^2 + (1-p)^2$

Proba de s'arrêter au k^e tour:

$$\mathbb{P}(k) = (p^2 + (1-p)^2)^k \times 2p(1-p)$$

$$= (2p^2 + 1 - 2p)^k \times 2p(1-p)$$

$$\sum_{k=0}^{+\infty} \mathbb{P}(k) = 2p(1-p) \times \frac{1}{1 - (2p^2 + 1 - 2p)} = \frac{2p(1-p)}{2p - 2p^2} = 1$$

(b) Proba de renvoyer $1 = \sum_{k=0}^{+\infty} (p^2 + (1-p)^2)^k \times p(1-p) = \frac{1}{2}$

= Proba de renvoyer 0

② $p = \sum_{k=1}^{+\infty} a_k 2^{-k}$

On tire à pile ou face jusqu'à obtenir le 1e face, disons au k^e coup, alors

| si $a_k = 0$ alors on renvoie 1
| si $a_k = 1$ alors on renvoie 0

$$P(\text{renvoyer } 0) = \sum_{k=1}^{+\infty} \frac{1}{2^k} \delta_{a_k=1} = \sum_{k=1}^{+\infty} \frac{1}{2^k} a_k = p$$

$$P(\text{renvoyer } 1) = \sum_{\substack{k=1 \\ \text{t.q. } a_k=0}}^{+\infty} \frac{1}{2^k} = 1-p.$$

③ (a) On enchaîne 2 tirages: $00 \to 0$, $01 \to 1$, $11 \to 3$, $10 \to 2$

(b) on enchaîne n tirages et on associe une valeur à chaque issue.

(c) on fait 3 tirages:

$000 \to 0$
$001 \to 1$
$011 \to 2$
$111 \to 3$
$100 \to 4$
$101 \to 5$
$010 \to$ on recommence
$110 \to$

$$Pr(i) = \frac{1/8}{6/8} = 1/6$$

(d) On tire des 5 valeurs, on en veut 7:
→ utiliser deux fois la fct → donne x et y
→ calculer $z = x + 5 * y \in [\![0,24]\!]$ uniformément
→ si $z \geq 21$, recommencer
→ sinon renvoyer $z \bmod 7$

Soit $k \in [\![0,6]\!]$,

$$P(k) = \frac{3/25}{21/25} = \frac{3}{21} = \frac{1}{7} \checkmark$$