TD d'algorithmique numéro 7
Daniel Hirschkoff et Natacha Portier
semaine du 15 au 19 novembre 1999

Question 1 ⋆ tas de Fibonacci ⋆

Tas de Fibonacci - définitions

Un tas de Fibonacci est un ensemble d'arbres qui respectent la *propriété de tas* (la clef d'un père est plus petite que la clef de ses fils). On associe à un tas T et à l'un quelconque de ses nœuds x les objets suivants :

- $min[T]$: un pointeur sur la plus petite racine de T (conséquemment, sur le nœud de plus petite clef) ;
- $n[T]$: nombre de nœuds dans T ;
- $a[T]$: nombre d'arbres dans T ;
- $p[x]$: père du nœud x ;
- $fils[x]$: *un des fils* de x (eh oui, cf ci-dessous) ;
- $gauche[x]$ et $droit[x]$: les fils d'un père donné sont reliés entre eux par une double liste circulaire chaînée : c'est pour cette raison que le père n'a besoin de ne pointer que vers l'un de ses fils ; les racines des arbres sont également reliées par une telle liste doublement chaînée.
- $degre[x]$: nombre de fils du nœud x ;
- $marque[x]$: indique si le nœud est marqué ou pas ; pour le moment, admettez que l'on peut marquer des nœuds, on verra à quoi ça sert dans la dernière partie ;
- $m[T]$: nombre de nœuds marqués dans T.

Vous pouvez dessiner un arbre d'un tas T pour comprendre comment les nœuds sont reliés entre eux.

Analyse amortie - rappels

Nous allons effectuer une analyse amortie à l'aide de la méthode des potentiels ; on définit pour cela la fonction potentiel

$$\phi(T) = a(T) + 2m(T)\,.$$

Lors du calcul des coûts amortis, on utilisera la grandeur $D(n)$, qui désigne le degré maximal d'un nœud pour un tas de Fibonacci à n nœuds.

Question 1.1 ⋆ arbres binomiaux non triés ⋆

Un ABNT (arbre binomial non trié) est une structure de données U_k, définie de la manière suivante :

- pour tout k, U_k est un tas ;

– U_0 est l'arbre à un nœud ;

– U_k est constitué à partir de deux arbres U_{k-1} en insérant la racine de l'un comme fils de la racine de l'autre.

Dessinez U_1, U_2, U_3, et montrez que :

1. U_k contient 2^k nœuds ;
2. la hauteur de U_k est k ;
3. U_k comporte C_k^i nœuds à la profondeur i pour $i \in [0 \ldots k]$;
4. la racine de U_k est de degré k, et ce degré est supérieur à celui de n'importe quel nœud.

Donner un majorant de $D(n)$ (cf plus haut) si tous les arbres constituant un tas de Fibonacci sont des ABNT.

Question 1.2 ⋆ algorithmes «simples» ⋆

Les tas de Fibonacci considérés sont des collections d'ABNT.

Les algorithmes d'insertion dans un tas de Fibonnaci et d'union de deux tas de Fibonacci sont évidents : on insère la nouvelle clef comme un arbre supplémentaire à un seul nœud, et on met côte à côte les deux tas pour faire l'union (en mettant à jour de manière évidente le pointeur sur la clef min à chaque fois).

On veut ensuite extraire du tas le nœud de clef minimum ; au cours de cette étape, on s'attache à ***consolider*** le tas obtenu après extraction, ce qui consiste en la construction progressive d'ABNT afin de limiter le nombre des racines du tas. L'idée est de regrouper les arbres pour fabriquer un tas tel que toutes les racines aient un degré distinct.

Écrire la fonction d'extraction du minimum, ainsi que la fonction de consolidation de l'arbre (pour l'instant, on laisse le marquage des différents nœuds de l'arbre inchangé). Évaluer les différents coûts.

Question 1.3 ⋆ diminution de clef ⋆

On veut maintenant diminuer la valeur d'une clef dans le tas. Les arbres du tas ne sont plus nécessairement des ABNT. On veut cependant que la borne supérieure $D(n)$ sur le degré d'un noeud quelconque d'un tas de Fibonacci à n noeuds vaille $O(\lg n)$. Prouver l'existence de cette borne permer d'inférer que l'extraction du minimum et la suppression s'exécutent dans un temps amorti en $O(\lg n)$.

On donne d'abord les algorithmes de diminution d'une clef et de suppression, puis dans les questions suivantes on montre la borne.

On utilise le marquage pour obtenir les bornes désirées. On veut qu'un nœud x ayant vécu les événements ci-dessous redevienne racine :

– à un moment, x a été racine,

- puis x a été relié à un autre nœud,
- puis deux fils de x lui ont été retirés par deux coupes.

Écrire l'algorithme de diminution de clef, et en évaluer le coût amorti. Comment comprendre dès lors la définition du potentiel pour les tas de Fibonacci ?

Comment définit-on ensuite l'algorithme de suppression de clef d'un tas de Fibonacci ?

Question 1.4 ⋆ degré maximum ⋆

La procédure de la question précédente ne maintient pas à tous les coups la propriété d´ABNT pour les arbres constituant le tas. On veut donc majorer le degré de tout nœud d'un tas de Fibonacci dans le cas général.

Soit x un nœud dans un tas de Fibonacci, et soit $k = degre[x]$. On apelle $y_1, y_2, \ldots, y_k$ les fils de x dans l'ordre où ils ont été créés. Montrer que $degre[y_1] \geq 0$ et pour $i \geq 2$, $degre[y_i] \geq i - 2$.

Question 1.5

On note F_n le n^{e} terme de la suite de Fibonacci, avec $F_0 = 0$ et $F_1 = 1$. Montrer que pour tout nœud x de degré k, $taille(x) \geq F_{k+2} \geq \phi^k$, où $\phi = (1+\sqrt{5})/2$ et $taille(x)$ désigne le nombre de nœuds du sous-arbre de racine x (x y compris).

Conclure en majorant le degré maximal $D(n)$ d'un nœud dans un tas de Fibonacci.