

ALGO1 – Tas binomiaux

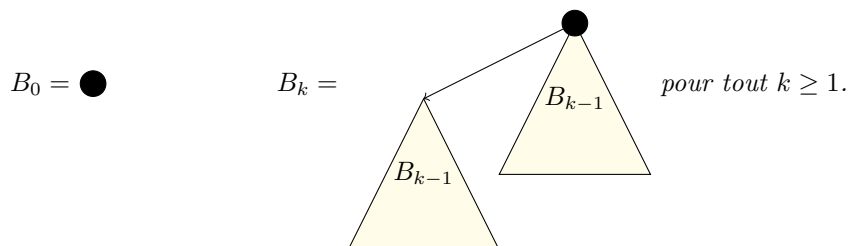
François Schwarzenruber

September 15, 2020

	Tas binaire	Tas binomiaux
enfiler	$O(\log n)$	$O(\log n)$
défiler	$O(\log n)$	$O(\log n)$
union	$O(n)$	$O(\log n)$

1 Arbres binomiaux

Définition 1 Les arbres binomiaux B_0, B_1, \dots sont définis par récurrence par :



Proposition 2

1. Il y a 2^k nœuds dans B_k .
2. La hauteur de B_k est k .
3. Il y a $\binom{k}{i}$ à la profondeur i .
4. La racine de B_k est d'arité k .

DÉMONSTRATION.

1. Par récurrence.
2. Par récurrence.
3. Par récurrence. On a $\binom{k-1}{i-1} + \binom{k-1}{i} = \binom{k}{i}$.
4. Par récurrence.

■

Corollaire 3 L'arité d'un arbre binomial à n éléments est $O(\log n)$.

DÉMONSTRATION. n est une puissance de deux : 2^k et la hauteur est $k = \log(n)$. ■

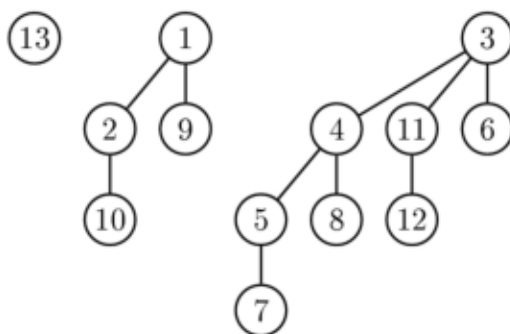
2 Définition d'un tas binomial

Définition 4 Un tas binomial H est un ensemble d'arbres binomiaux étiquetés tel que :

- chaque arbre de H vérifie la propriété de tas, i.e. chaque nœud est plus prioritaire que ses enfants ;
- chaque B_k apparaît au plus une fois dans H .

Proposition 5 *Un tas binomial à n éléments contient au plus $O(\log n)$ arbres.*

DÉMONSTRATION.



Il y a 13 éléments et $13 = 1 + 2^2 + 2^3$. Le nombre d'arbres est le nombre de chiffres qu'il faut pour écrire le nombre n , c'est un $O(\log n)$. ■

3 Implémentation

Liste chaînée sur les racines par degrés croissants.

En chaque nœud : on stocke le degré et pointeurs vers le parent, premier fils, frère droit



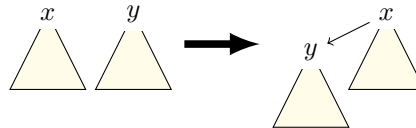
4 Opérations

Créer un tas vide. Créer une liste vide.

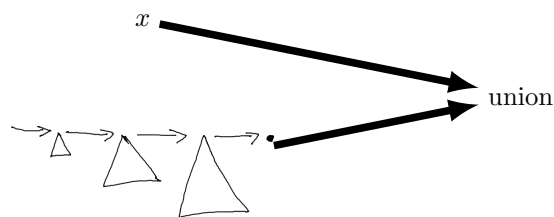
Trouver l'élément prioritaire. Parcourir la liste des racines.

Union de deux tas binomiaux.

1. On fusionne les deux listes par ordre de degré croissant.
2. On colle les arbres de même degré :

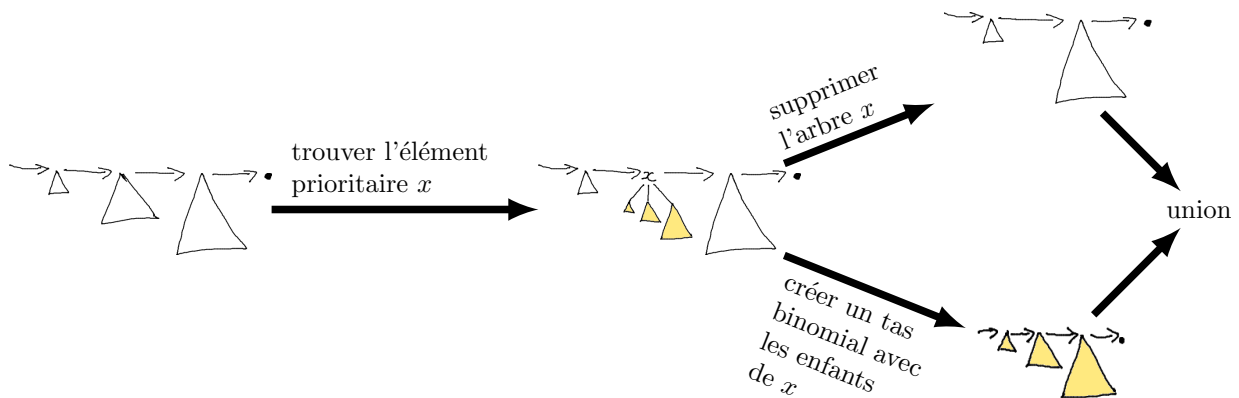


Insérer l'élément x .



- Ajouter l'arbre-racine x au début de liste
- Union 2.

Défiler l'élément prioritaire.



5 Notes bibliographiques

Les tas binomiaux [Vui78] permettent l'union en $O(\log n)$.

References

[Vui78] Jean Vuillemin. A data structure for manipulating priority queues. *Commun. ACM*, 21(4):309–315, 1978.