Arbres, tas binaires et tas de Fibonacci

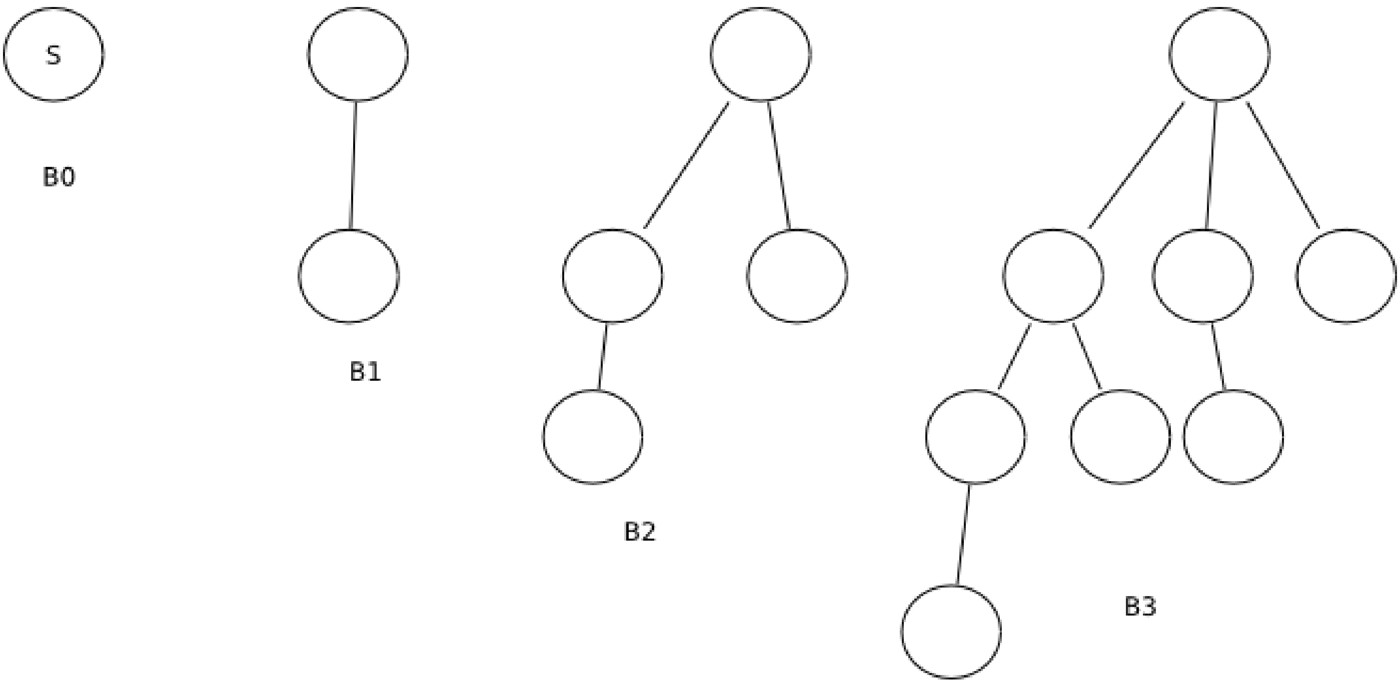
1 Arbres binômiaux (Binomial Trees)

1.1 Définition

Un BTree est défini récursivement de la façon suivante :

- B_0 consiste en un sommet
- B_k est constitué à partir de deux arbres binomiaux
- B_{k-1} est lié de la façon suivante : la racine de l'un est le fils le plus à gauche de l'autre.

Exemple :



1.2 Propriétés

1. Un arbre B_k possède 2^k noeuds
2. La hauteur est k
3. Il y a exactement $\begin{pmatrix} k \\ i \end{pmatrix}$ noeuds à la profondeur i, pour $i \in [1, k]$.
4. La racine de l'arbre a le plus grand degré $i = k$
5. Si on numérote les fils de gauche à droite, le fils i est la racine de B_i.

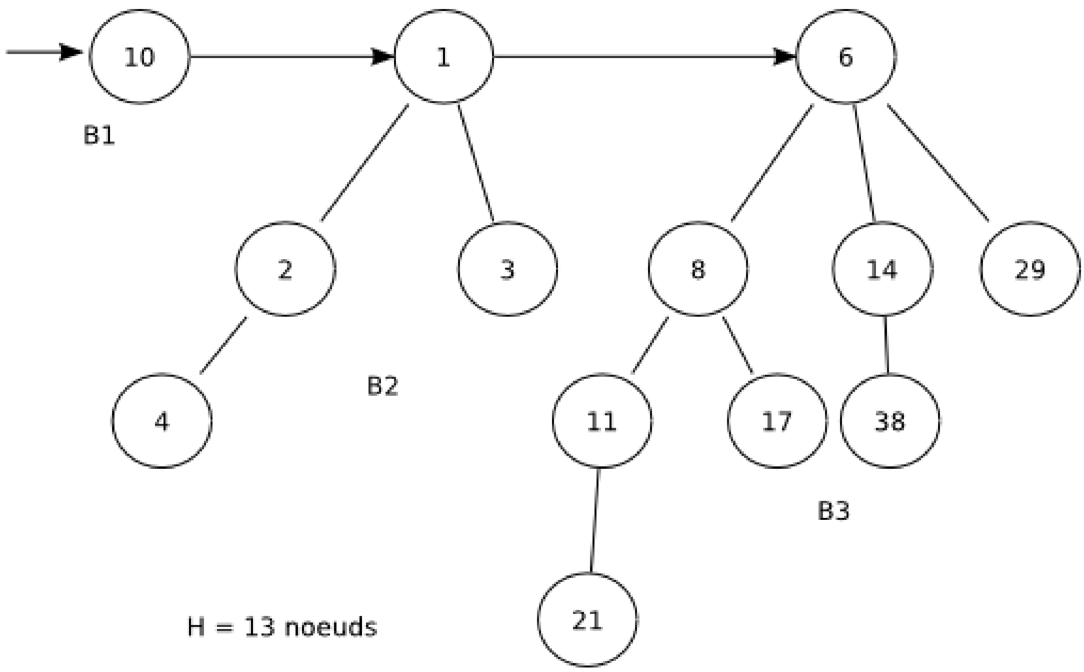
2 Tas binomiaux (Binomial Heaps)

2.1 Définition

Un BH est un ensemble d'arbres binômiaux.

– Chaque BT dans H vérifie la propriété du tas, c'est à dire la clé d'un noeud est inférieure ou égale aux clés de ses fils.
– un tas contenant n noeuds contient au plus $\lfloor log(n) \rfloor + 1$ BT. un BT d'un degré donné apparait au plus une fois.

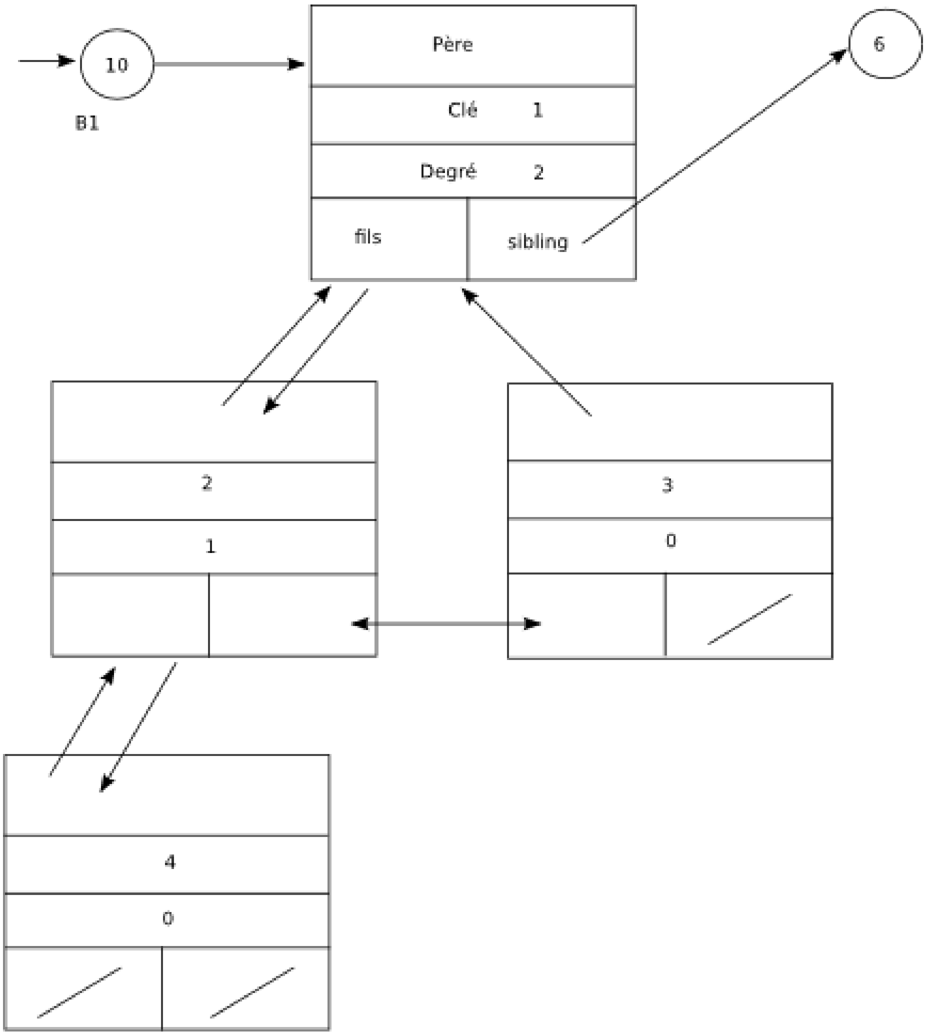
Exemple :



Les racines binomiales sont reliées en liste (appelée liste de racines), selon l'ordre croissant de degrés.

2.2 Représentation d'un noeud

A partir de l'exemple précédent :



2.3 Opérations

Opérations	Binomial Heap	Binary Heap	Fibonacci Heap
MakeHeap	$O(1)$	$O(1)$	$O(1)$
Insert	$O(log(n))$	$O(log(n))$	$O(1)$
Minimum	$O(log(n))$	$O(1)$	$O(1)$
ExtraitMin	$O(log(n))$	$O(log(n))$	$O(log(n))$
DecreaseKey	$O(log(n))$	$O(log(n))$	$O(1)$
Delete	$O(log(n))$	$O(log(n))$	$O(log(n))$
Union	$O(log(n))$	$O(n)$	$O(1)$

2.3.1 L'opération d'union

Le problème est que l'on dispose potentiellement de plusieurs BT de même degré, donc on ne respecte plus les propriétés. Il faut reconstruire, afin de corriger.

L'algorithme BH-Union cosiste à parcourir la liste des racines et supprimer (mettre à jour) les racines des BT en double.

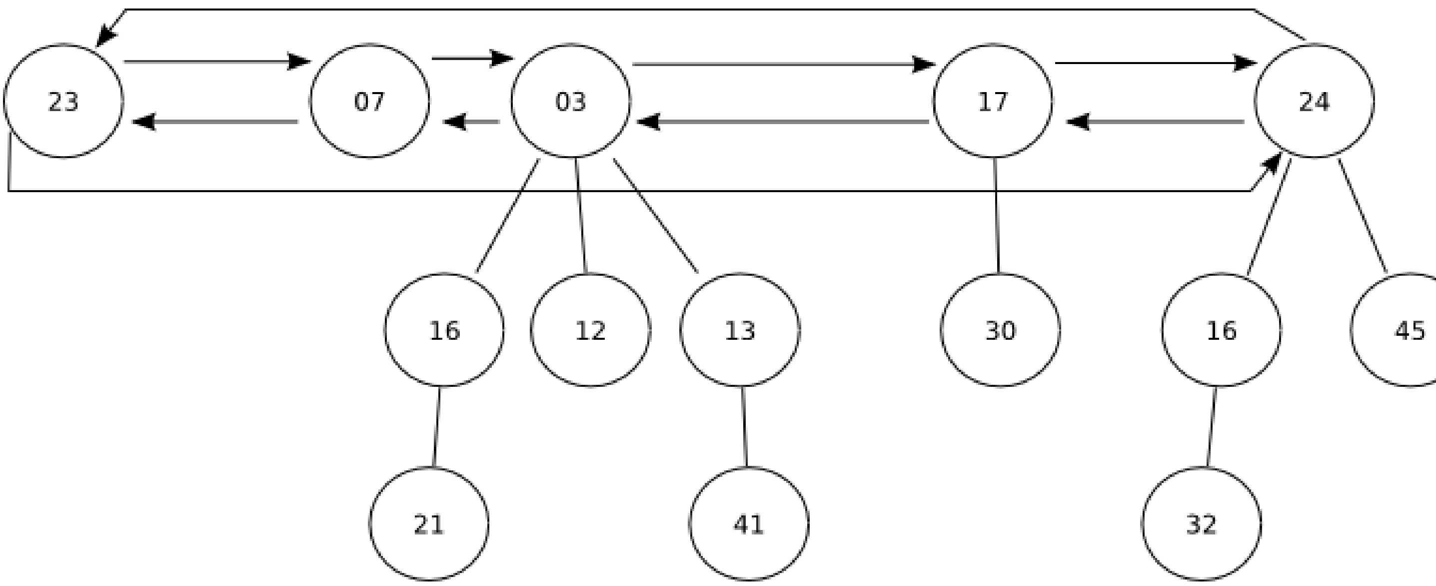
Le coût de l'algorithme est dans le pire des cas $O(log(n))$, n étant la somme du nombre de noeuds des deux tas à fusionner.

3 Tas de Fibonacci (Fibonacci Heaps)

Utilisé dans MST et Dijkstra.

3.1 Définition

FH est une collection d'arbres dotés de la propriété de tas ; ces arbres ne sont pas obligatoirement binomiaux).



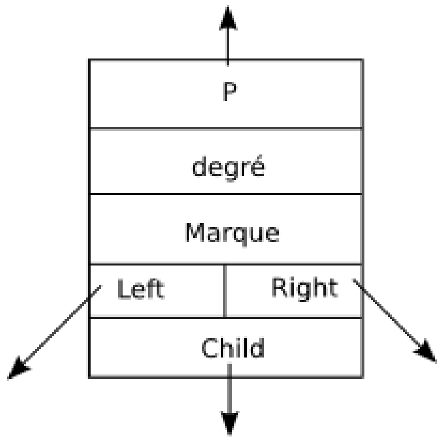
Ci dessus, exemple d'uns FH avec 5 tas et 14 noeuds.

- Les racines des arbres sont connectées en liste circulaire doublement chaînée.
- Le FH est accessible par le pointeur $min(H)$
- L'ordre des arbres est arbitraire

– On stocke le nombre de noeuds courants de H dans n[H].

3.2 Représentation d'un noeud

On peut représenter comme suit :



“Marque” est un booléen pour indiquer si x a perdu un fils depuis la dernière fois que x est devenu le fils d'un autre noeud.