

Tas

Benjamin Monmege



Structure de données abstraite : file de priorité (minimum)

- Test du vide de la file de priorité
- Insertion d'un nouvel élément avec une priorité donnée
- Recherche de l'élément de priorité minimum
- Extraction de l'élément de priorité minimum

Interface :

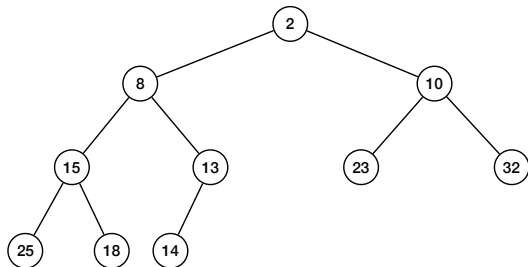
```
heap.isempty()           # true or false
heap.insert(element, priority) # modifies heap
heap.minimum()          # returns element
heap.extract_minimum()  # remove minimum priority
```

Utilisation de tas binaires

Arbre binaire *parfait* étiqueté par les priorités tel que pour tout nœud p d'étiquette x :

- toutes les étiquettes des enfants de p sont supérieures à x

Parfait signifie que tous les niveaux sont complets, sauf le dernier rempli de gauche à droite



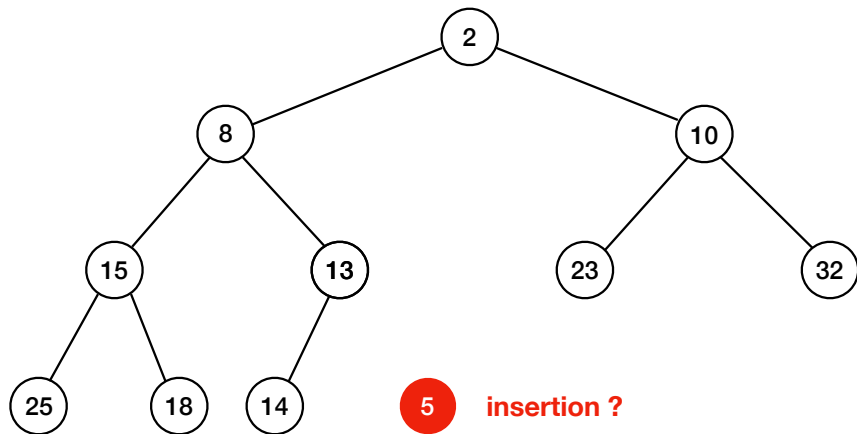
La hauteur d'un tas binaire à n nœuds est égale à $\lceil \log_2(n) + 1 \rceil$.

- Test du vide de l'arbre binaire. . .

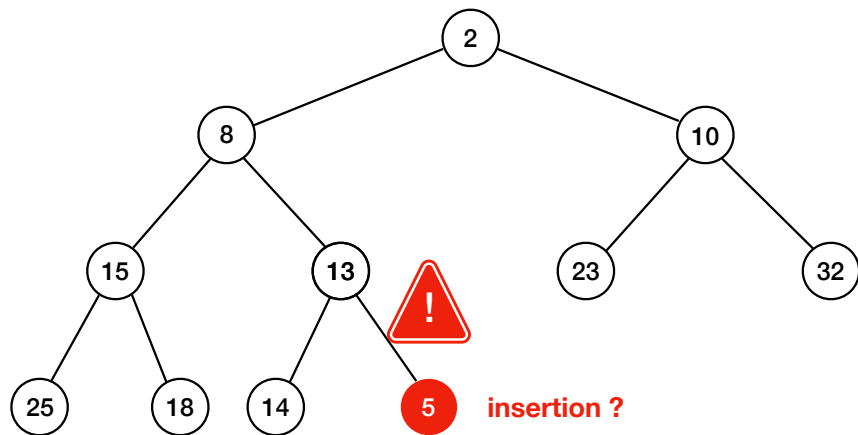
- Test du vide de l'arbre binaire. . .

Complexité : $O(1)$

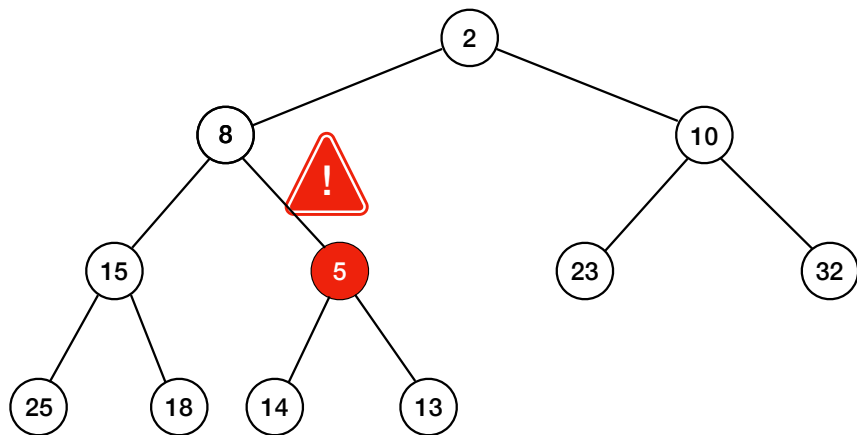
Insertion d'une nouvelle priorité



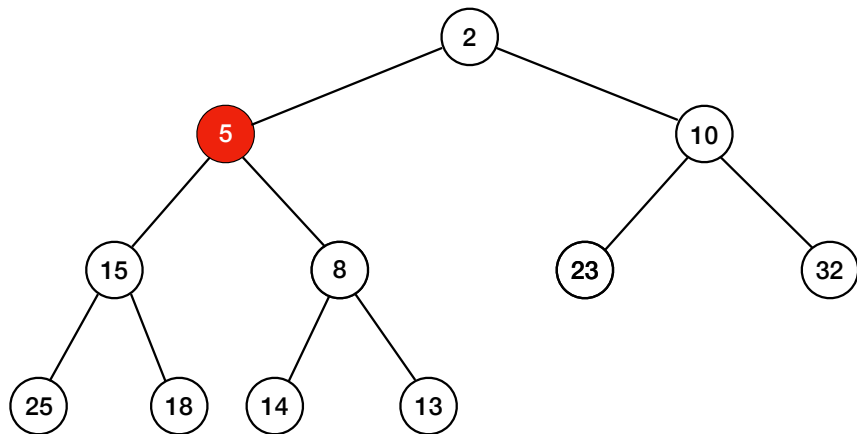
Insertion d'une nouvelle priorité



Insertion d'une nouvelle priorité



Insertion d'une nouvelle priorité



Insertion d'une nouvelle priorité

- Insérer à la prochaine place libre dans l'arbre qui doit rester parfait,
- puis faire remonter le long de la branche jusqu'à satisfaire la contrainte sur les priorités

Insertion d'une nouvelle priorité

- Insérer à la prochaine place libre dans l'arbre qui doit rester parfait,
- puis faire remonter le long de la branche jusqu'à satisfaire la contrainte sur les priorités

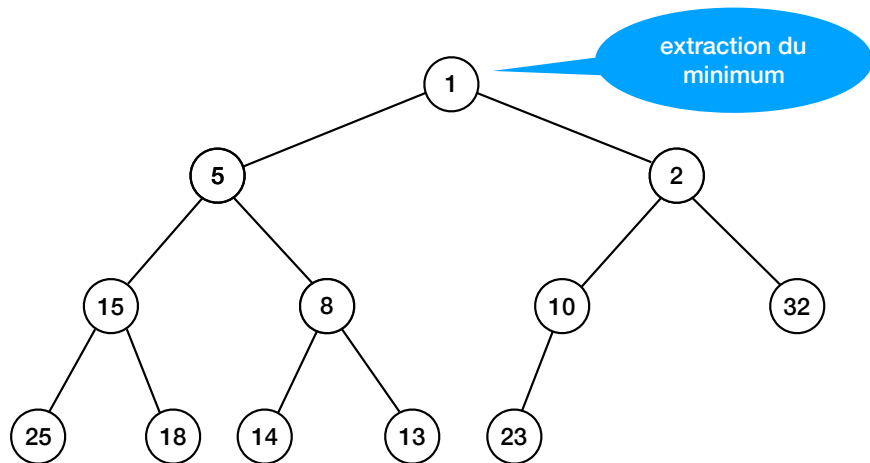
Complexité : $O(\log n)$

- La priorité minimum est à la racine de l'arbre

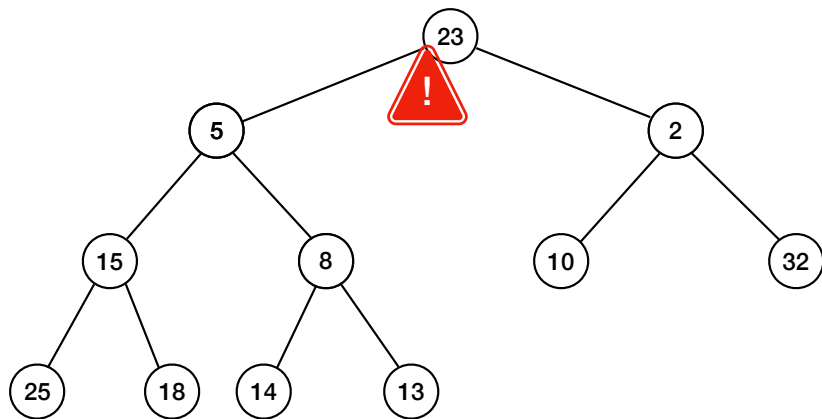
- La priorité minimum est à la racine de l'arbre

Complexité : $O(1)$

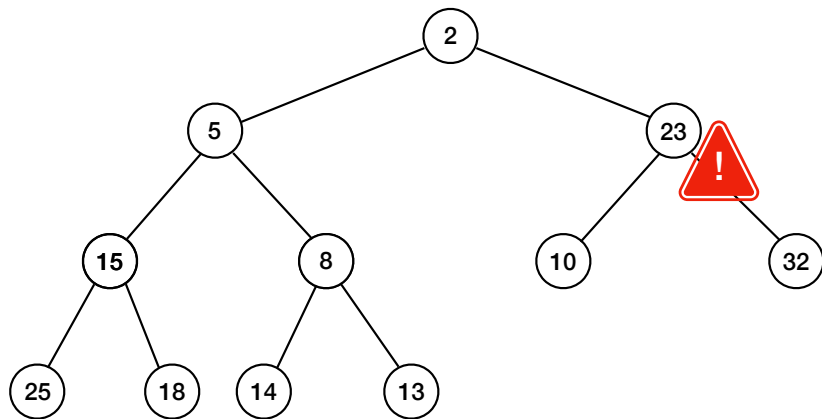
Extraction de l'élément de priorité minimum



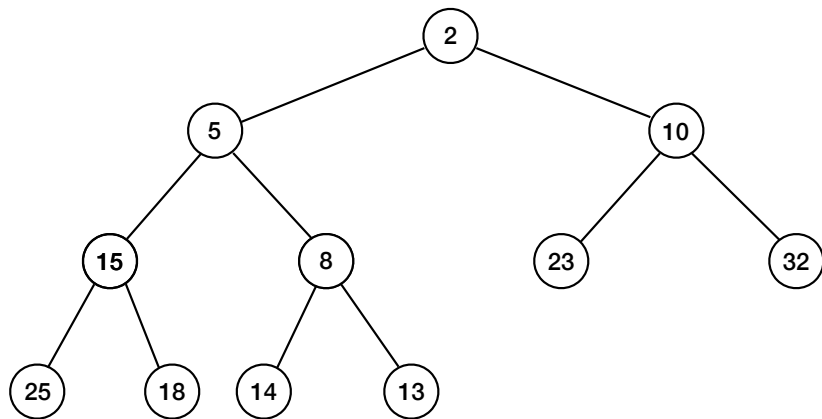
Extraction de l'élément de priorité minimum



Extraction de l'élément de priorité minimum



Extraction de l'élément de priorité minimum



Extraction de l'élément de priorité minimum

- Remplacer la racine de l'arbre par le dernier élément du niveau le plus bas (pour que l'arbre reste parfait)
- Faire descendre la priorité le long de la branche qui mène à l'élément de priorité minimum, jusqu'à satisfaire la contrainte sur les priorités

Extraction de l'élément de priorité minimum

- Remplacer la racine de l'arbre par le dernier élément du niveau le plus bas (pour que l'arbre reste parfait)
- Faire descendre la priorité le long de la branche qui mène à l'élément de priorité minimum, jusqu'à satisfaire la contrainte sur les priorités

Complexité : $O(\log n)$