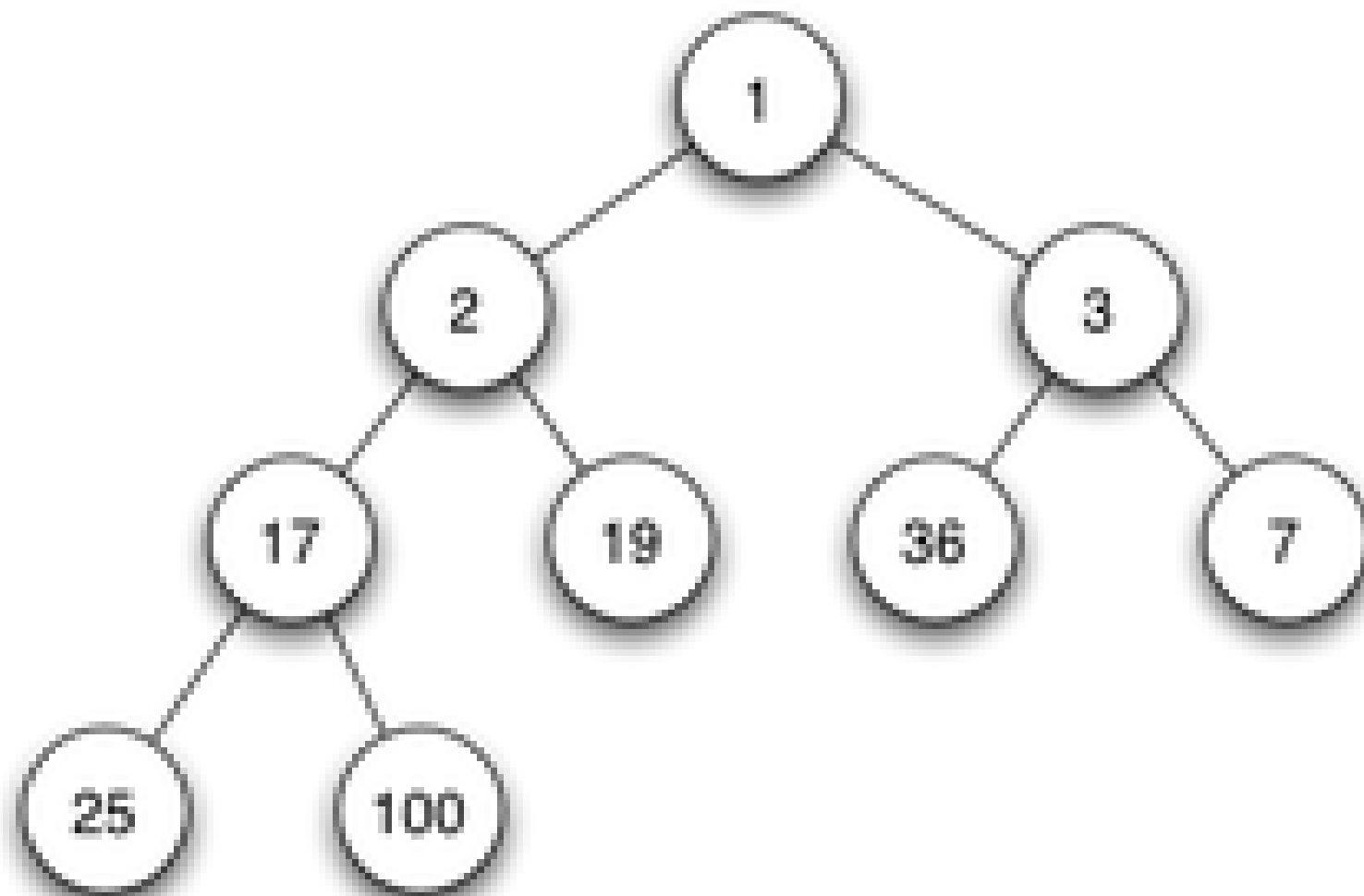




DEVOIR DE PROGRAMMATION

Hakim Baaloudj | Lydia Ronava





Tas priorité min

Implémentation basé sur un tableau

- On travaillent avec des indices
- Aller chercher le dernier élément coûte rien.
- Parcours avec indices

Implémentation avec une structure auto référencée

- On travaillent avec des références.
- Exactement temps logarithmique pour aller chercher le dernier élément, constant pour la racine, et $h(\text{noeud})$ pour tout autre nœud
- Parcours avec représentation binaire (slide suivant)

On veut insérer un 13

On calcul la représentation binaire de 9
(prochain indice ou insérer l'élément) [1,0,0,1]

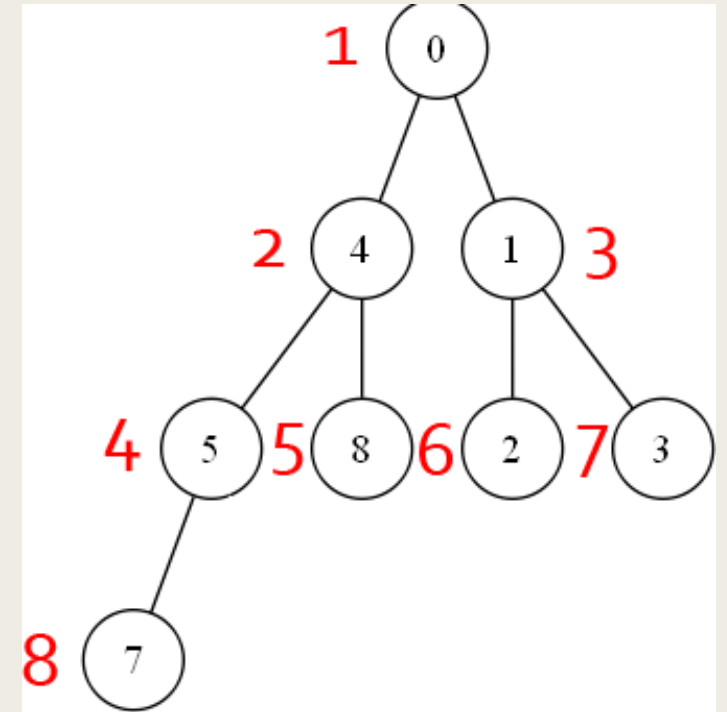
On ignore le premier chiffre qui représente la racine et

On respecte cet ordre

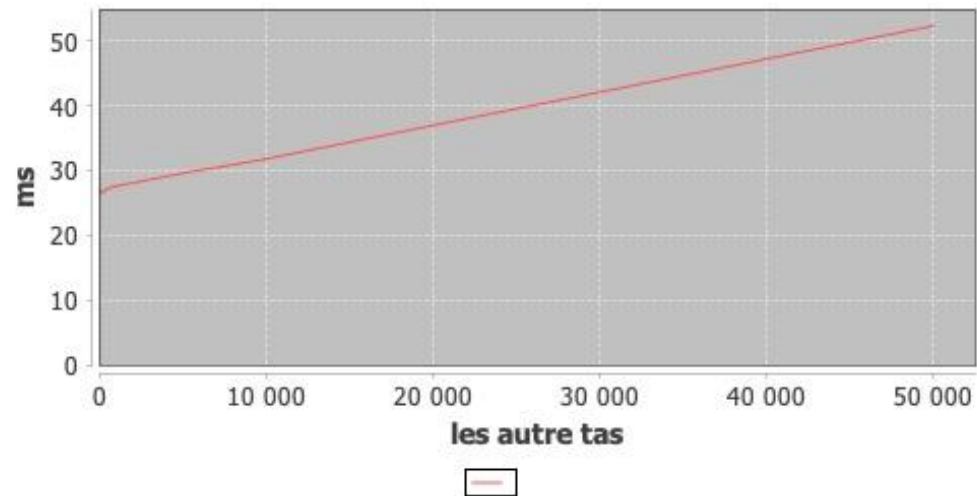
0 -> aller a gauche

1 -> aller a droite.

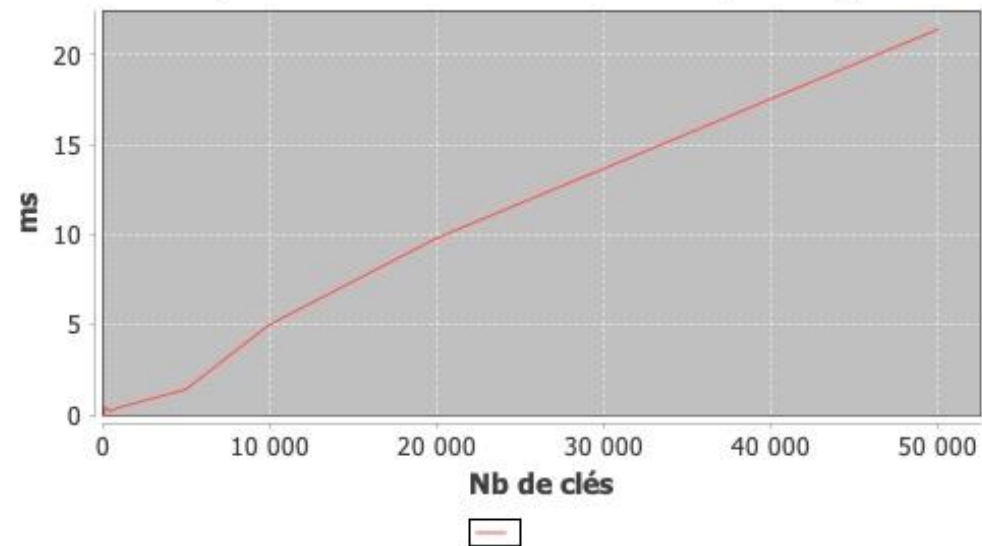
Pour insérer on doit aller deux fois a gauche, une fois a droite.



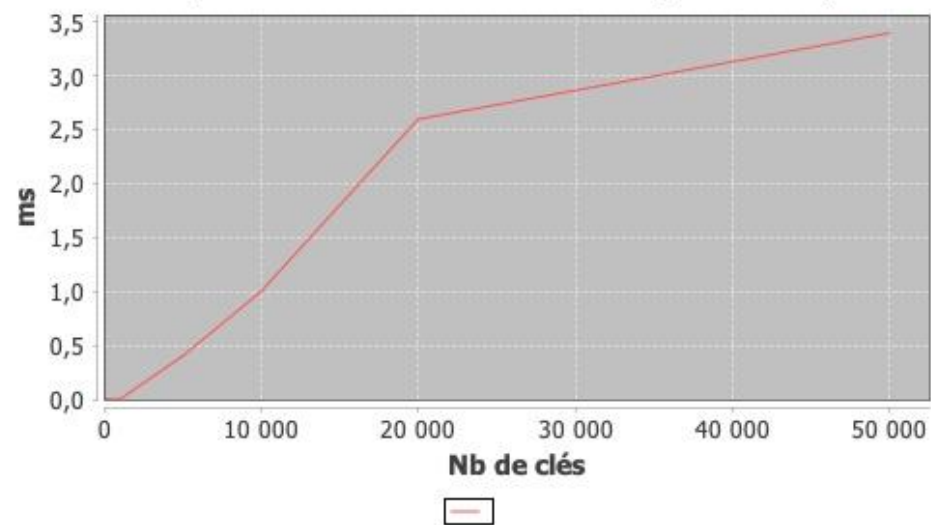
**Complex union tasMinTree avec un tas initial
20K clé**



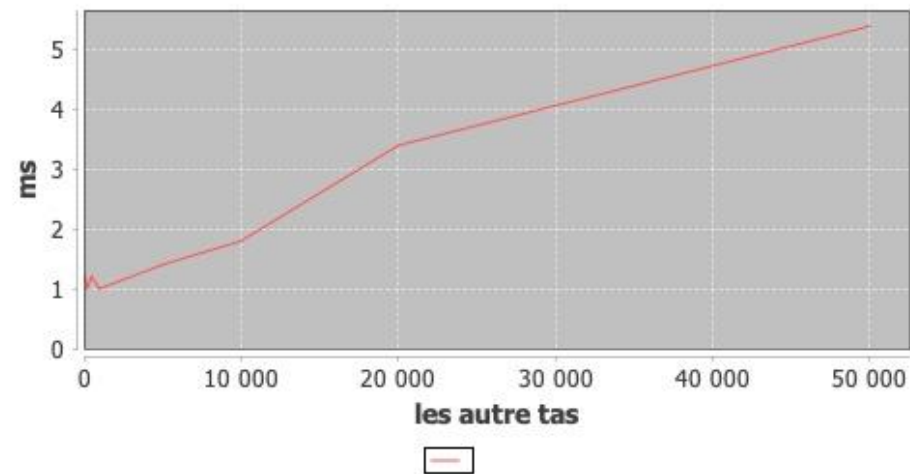
Complexité consIter Tasmin (arbre)



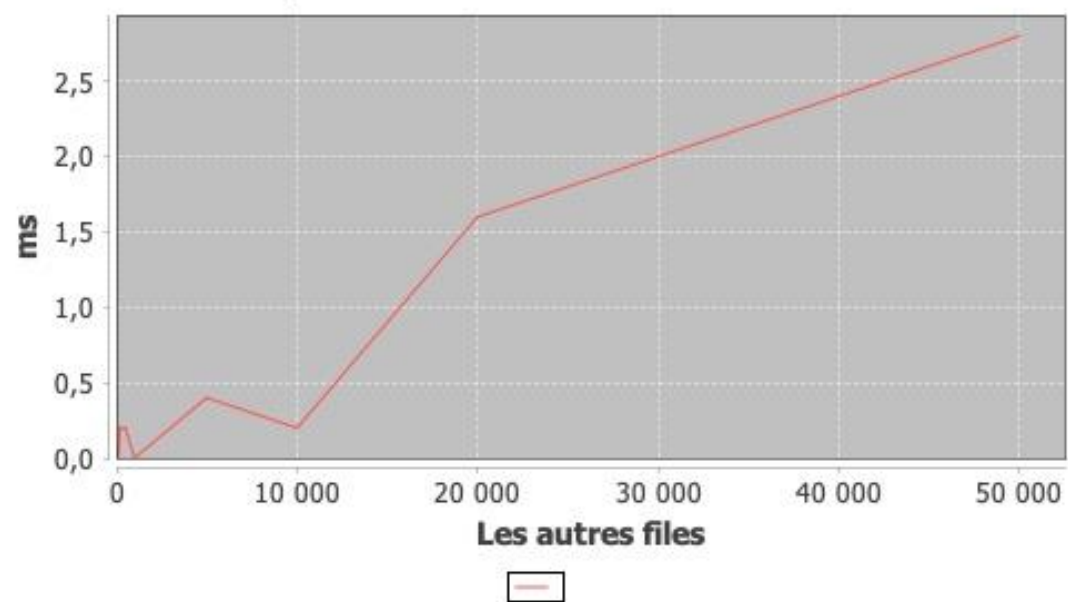
Complexité consIter Tasmin (tableau)



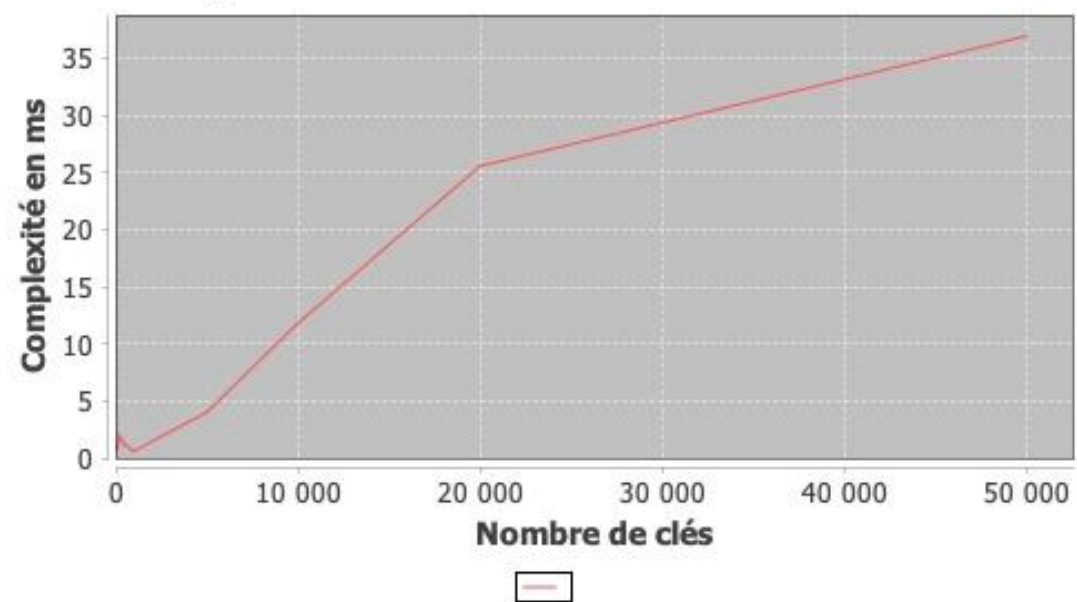
**Complex union tasMinArray avec un tas initial
20K clé**



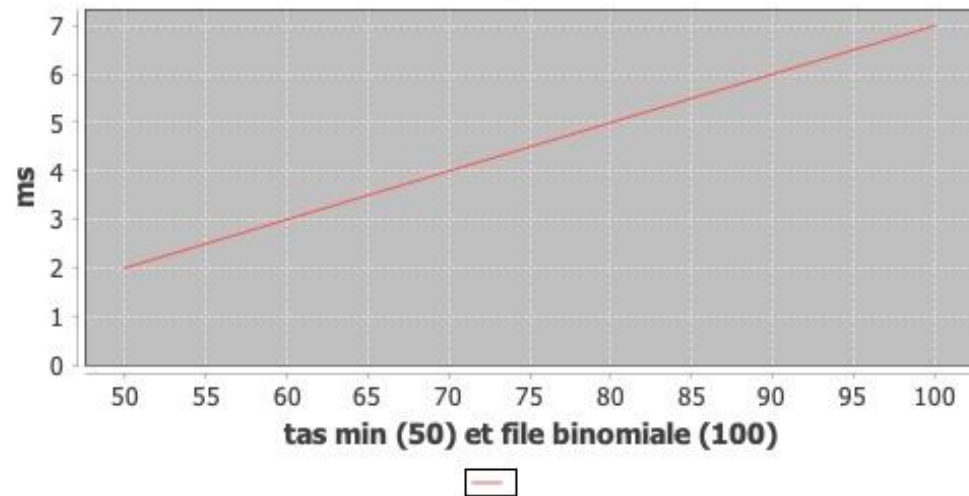
Complexité union FileBinomiale



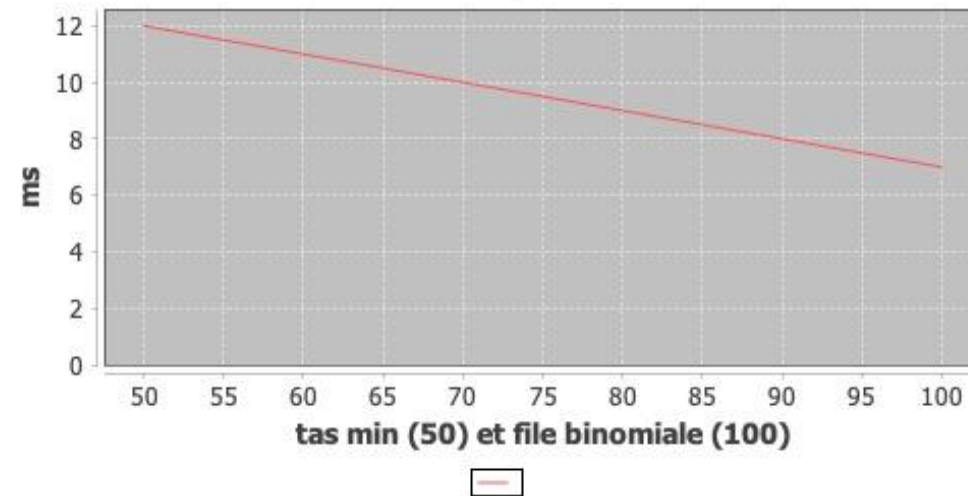
Complexité ConsIter sur FileBinomiale



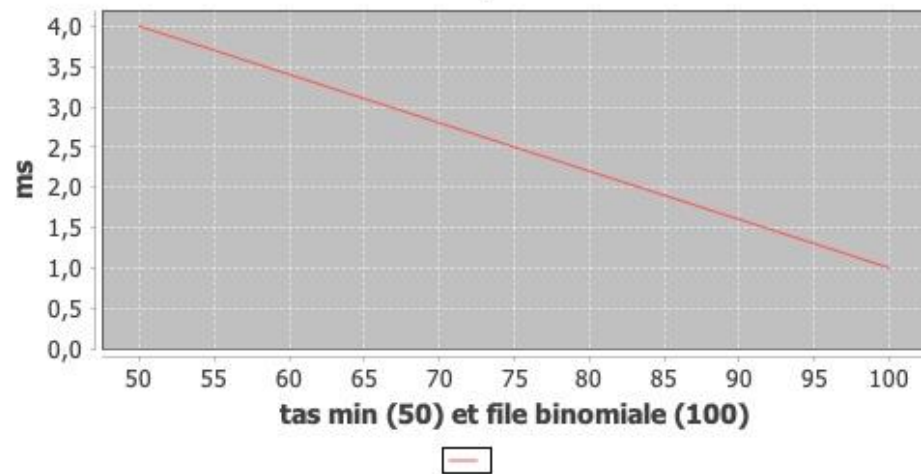
**temps d'execution ajout sur mot distinct
Shakespeare**



**temps d'execution supprMin sur mot distinct
Shakespeare**



**temps d'execution union sur mot distinct
Shakespeare**



**temps d'execution ConsIter sur mot distinct
Shakespeare**

