

# MU4IN500 - Algorithmique avancée

## Devoir de programmation

FLoria LIM 28706087

Myriam MABROUKI 28710344

Enseignants :

Antoine Genitrini, Mehdi Naima

Automne 2023

# Plan

- 1 Clé
- 2 Tas priorité min
- 3 File Binomiale
- 4 Arbre de Recherche
- 5 Étude expérimentale

# Plan

- 1 Clé
- 2 Tas priorité min
- 3 File Binomiale
- 4 Arbre de Recherche
- 5 Étude expérimentale

## Structure

- un tableau de taille 4 où chaque élément est un entier non signé de 32 bits

# Plan

- 1 Clé
- 2 **Tas priorité min**
- 3 File Binomiale
- 4 Arbre de Recherche
- 5 Étude expérimentale

# Tas priorité min

## Sous forme de tableau

- un `vector` de `Key` possédant l'ensemble des clés du tas
- un entier `size` qui représente le nombre de clés du tas

## Sous une forme arborescente

- un pointeur `value` de type `Key` qui représente la clé
- deux pointeurs de type `Heap_tree` qui représentent les fils gauche, `left` et droit `right`
- un entier `size` qui représente la taille de l'arbre

# Construction

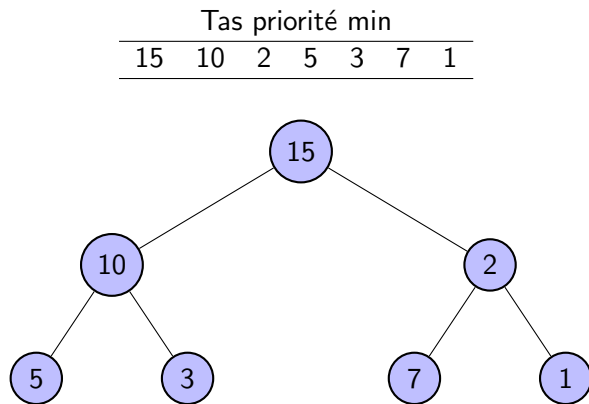


Figure: Exécution de construction

# Construction

Tas priorité min						
15	10	2	5	3	7	1

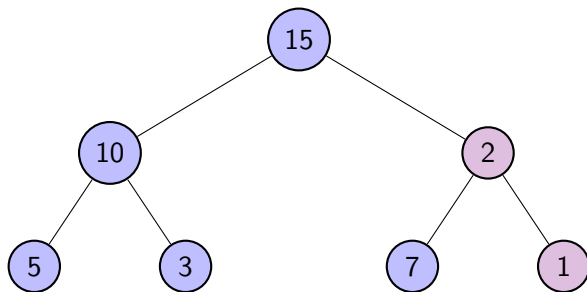


Figure: Exécution de construction



# Construction

Tas priorité min						
15	10	1	5	3	7	2

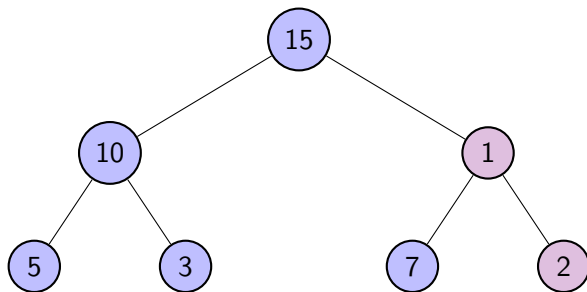


Figure: Exécution de construction

# Construction

Tas priorité min						
15	10	1	5	3	7	2

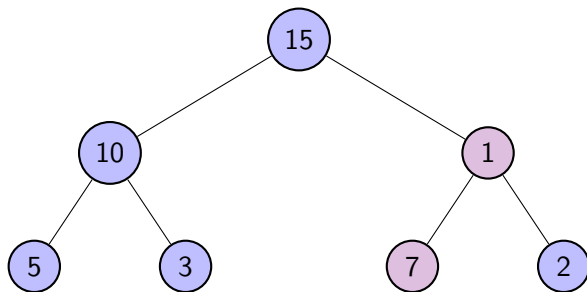


Figure: Exécution de construction

# Construction

Tas priorité min						
15	10	1	5	3	7	2

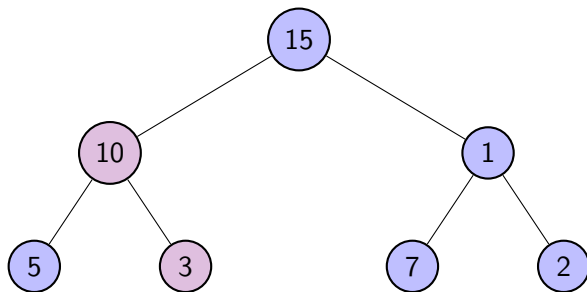


Figure: Exécution de construction

# Construction

Tas priorité min						
15	3	1	5	10	7	2

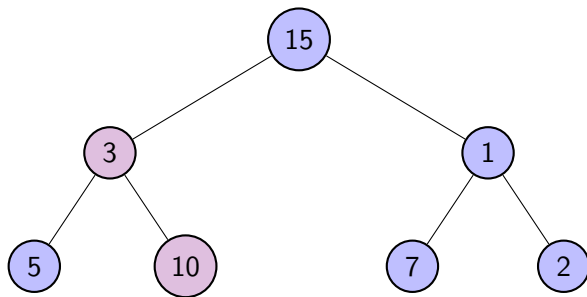


Figure: Exécution de construction

# Construction

Tas priorité min						
15	3	1	5	10	7	2

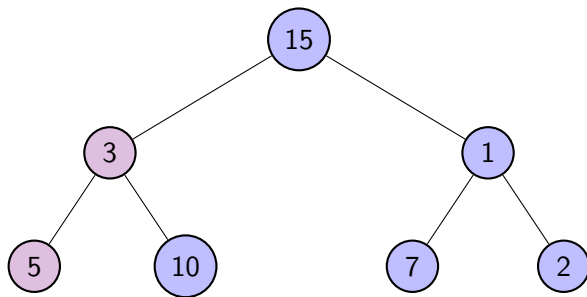


Figure: Exécution de construction

# Construction

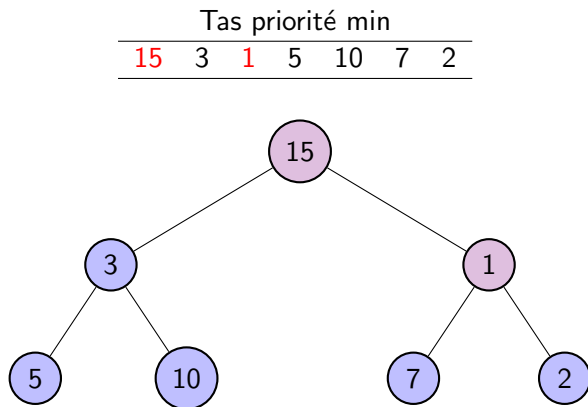


Figure: Exécution de construction

## Construction

## Tas priorité min

1 3 15 5 10 7 2

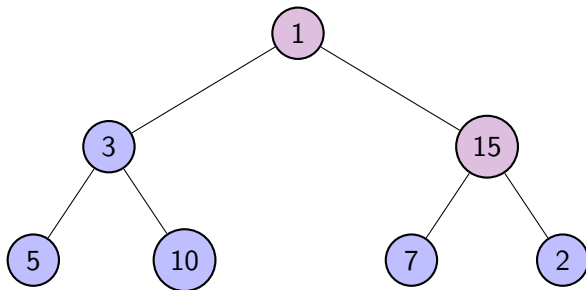


Figure: Exécution de construction

# Construction

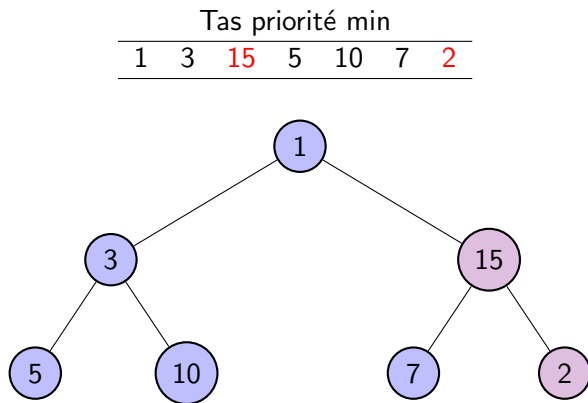


Figure: Exécution de construction



# Construction

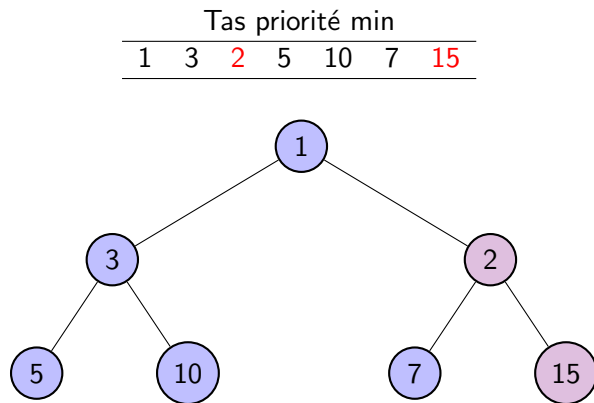


Figure: Exécution de construction

# Construction

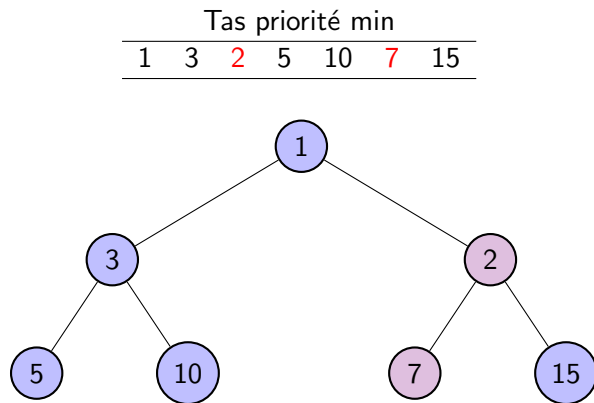


Figure: Exécution de construction

# Construction

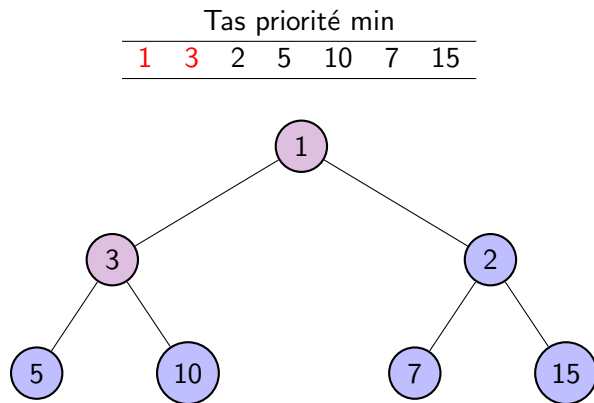


Figure: Exécution de construction

# Temps d'exécution de la création d'un tas min

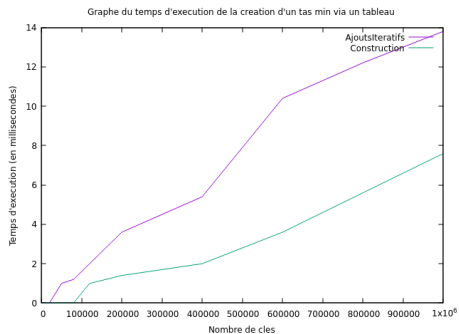


Figure: Sous la forme d'un tableau

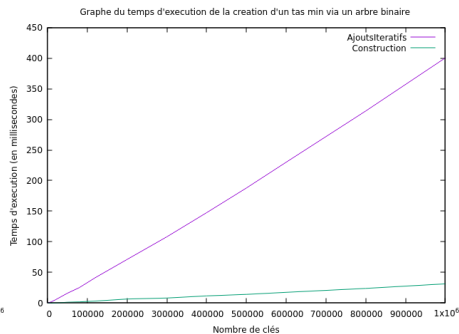


Figure: Sous une forme arborescente

Tas priorité min



Figure: Exécution de Union

Tas priorité min

2

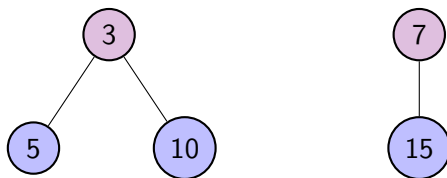


Figure: Exécution de Union

# Union

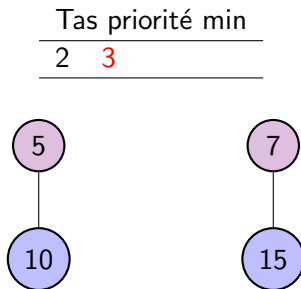


Figure: Exécution de Union

# Union

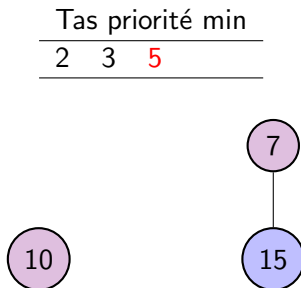


Figure: Exécution de Union



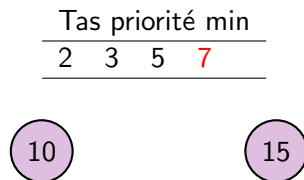


Figure: Exécution de Union

Tas priorité min

---

2 3 5 7 10

---

15

Figure: Exécution de Union

Tas priorité min					
2	3	5	7	10	15

Figure: Exécution de Union

# Temps d'exécution de l'union de deux tas min

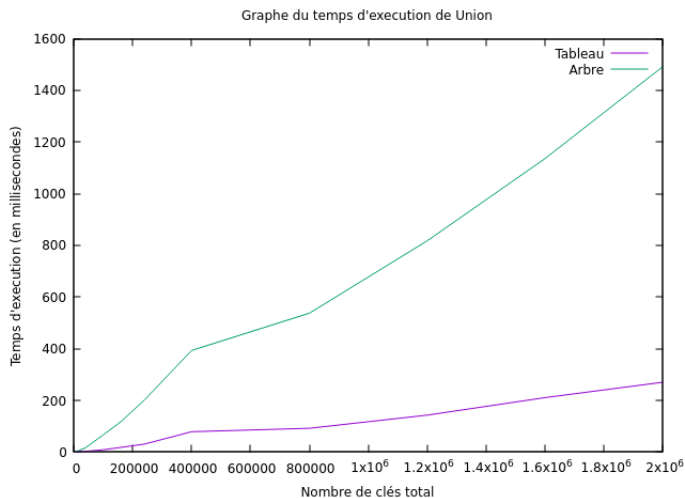


Figure: Union de deux tas min

# Plan

- 1 Clé
- 2 Tas priorité min
- 3 File Binomiale**
- 4 Arbre de Recherche
- 5 Étude expérimentale

## Tounois Binomial

- un vector de `TounoiBinomial` qui représente la forêt liée à la racine du tournois
- un entier, `size`, qui représente le nombre de noeuds du tournoi

## File binomiale

- un vector de `TounoiBinomial` qui représente la suite de tournois binomiaux dont est composée la file
- un entier, `size`, qui représente le nombre de noeuds de la file

# Temps d'exécution de la construction d'une file binomiale

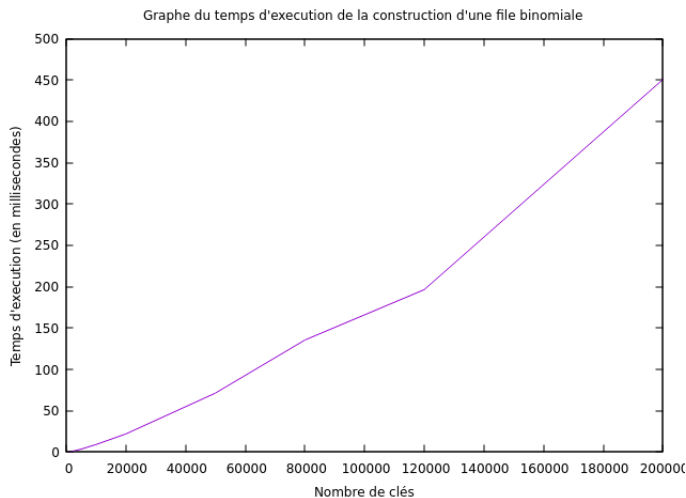


Figure: Construction d'une file binomiale

# Temps d'exécution de l'union de deux files binomiales

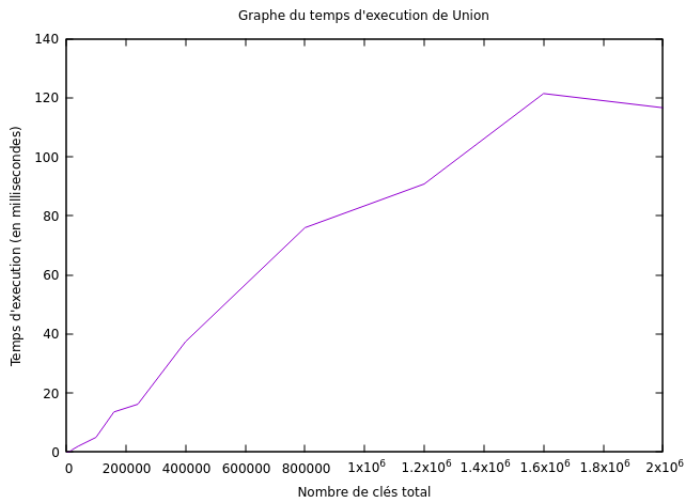


Figure: Union de deux files binomiales



# Plan

- 1 Clé
- 2 Tas priorité min
- 3 File Binomiale
- 4 Arbre de Recherche**
- 5 Étude expérimentale

## Structure

- un pointeur `key` de type `Key` qui représente le haché MD5 d'un mot
- une chaîne de caractères `value` où on y stocke le mot
- deux pointeurs de type `BinarySearchTree` qui représentent les fils gauche, `left` et droit `right`
- un entier, `size`, qui représente la taille de l'arbre

# Plan

- 1 Clé
- 2 Tas priorité min
- 3 File Binomiale
- 4 Arbre de Recherche
- 5 Étude expérimentale

# Temps d'exécution comparatif sur Ajout

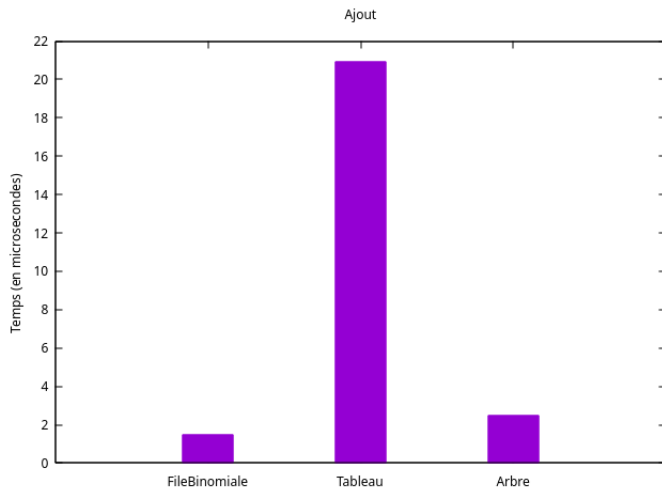


Figure: Ajout sur une file, un tas via un tableau et un tas via un arbre

# Temps d'exécution comparatif sur SupprMin

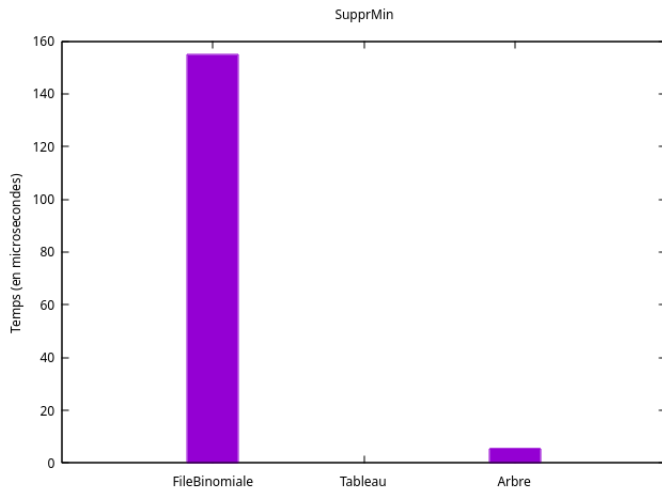


Figure: SupprMin sur une file, un tas via un tableau et un tas via un arbre

# Temps d'exécution comparatif sur Construction

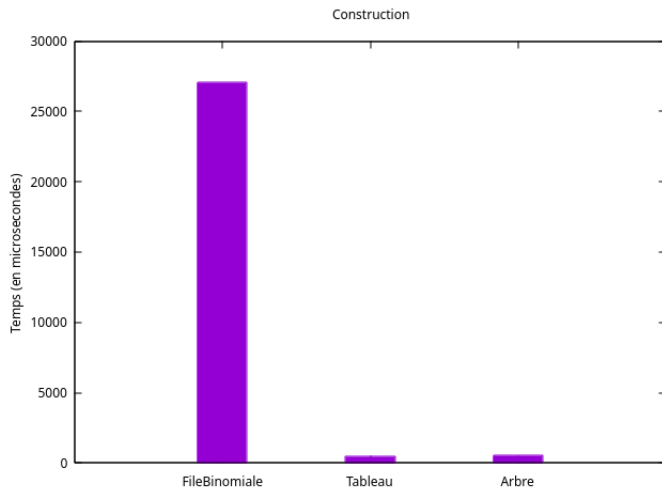


Figure: Construction sur une file, un tas via un tableau et un tas via un arbre

# Temps d'exécution comparatif sur Union

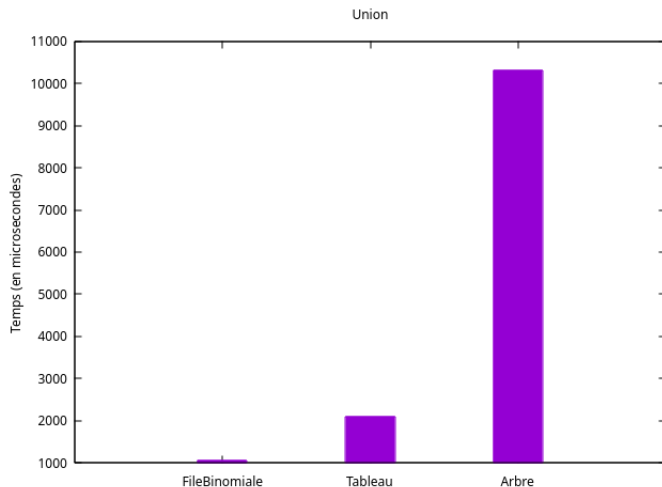


Figure: Union sur une file, un tas via un tableau et un tas via un arbre