



SORBONNE
UNIVERSITÉ

DEVOIR DE PROGRAMMATION ALGORITHMIQUE AVANCÉ

SHIYAO CHEN
ATSE YAPI

MU4IUN500
MASTER STL
2023-2024

SOMMAIRE

Partie 1 : Présentation	3
Structure de clé	4
Partie 2 : Tas priorité min	5
Structures de données	6
Complexités temporelles	7
Fonctions : Ajouts itératifs & Construction (arbre)	8
Fonction : Union (arbre)	8
Fonctions : Ajouts itératifs & Construction (tableau)	9
Fonction : Union (tableau)	9
Partie 3 : File Binomiale	10
Structures de données	11
Complexités temporelles	12
Fonction : Construction	13
Fonction : Union	14
Partie 4 : Fonction de hachage	15
Représentation	16
Partie 5 : Arbre de recherche	17
Structures de données	18
Primitives	19
Complexités temporelles	20
Fonction : Exists	21
Partie 6 : Etude Expérimentale	22
Structures de données	23
Comparaison graphique des complexité temporelles : Tas Min & File Binomiale	24
Fonction : Suppr Min	25
Fonction : Ajout	26
Fonction : Construction	27
Fonction : Union	28
Conclusion	29



PARTIE 1

Présentation



STRUCTURE DE CLÉ

ENTIER CODÉ SUR 128 BITS

```
typedef struct cle_entier{  
    uint32_t u1; // bit de poids faible  
    uint32_t u2;  
    uint32_t u3;  
    uint32_t u4; // bits de poids forts  
}Cle_entier;
```

LISTE DE CLÉS

```
typedef struct ce_cell{  
    Cle_entier* cle;  
    struct ce_cell* suiv;  
}Ce_cell;
```



PARTIE 2

LES TAS PRIORITÉ MIN



STRUCTURES DE DONNÉES

TAS PRIORITÉ MIN

VIA UN ARBRE BINAIRE

```
typedef struct tas_ab{  
    Cle_entier* cle;  
    struct tas_ab* fg;  
    struct tas_ab* fd;  
    int pfd;  
} Tas_Min_arbre;
```

VIA UN TABLEAU

```
typedef struct tas_min_tableau {  
    int capacite; //case possible  
    int taille; //taille réelle  
    //Cle_entier* lcle;  
    Cle_entier** tab;  
} Tas_Min_tableau;
```

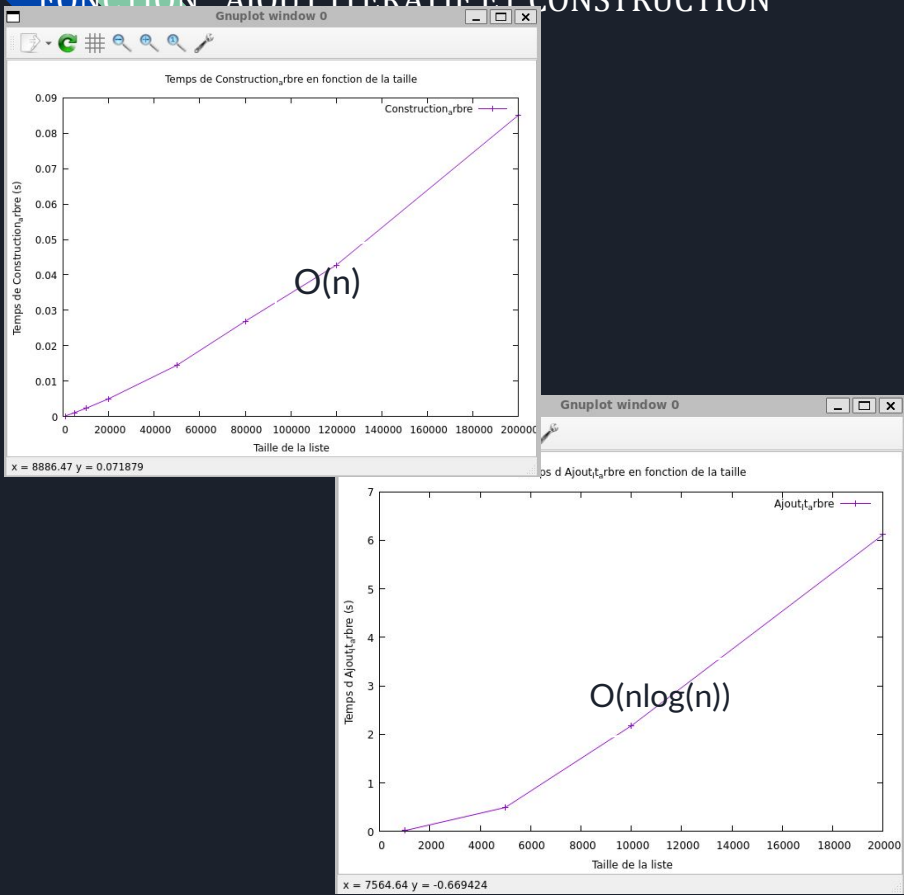


COMPLEXITÉS TEMPORELLES

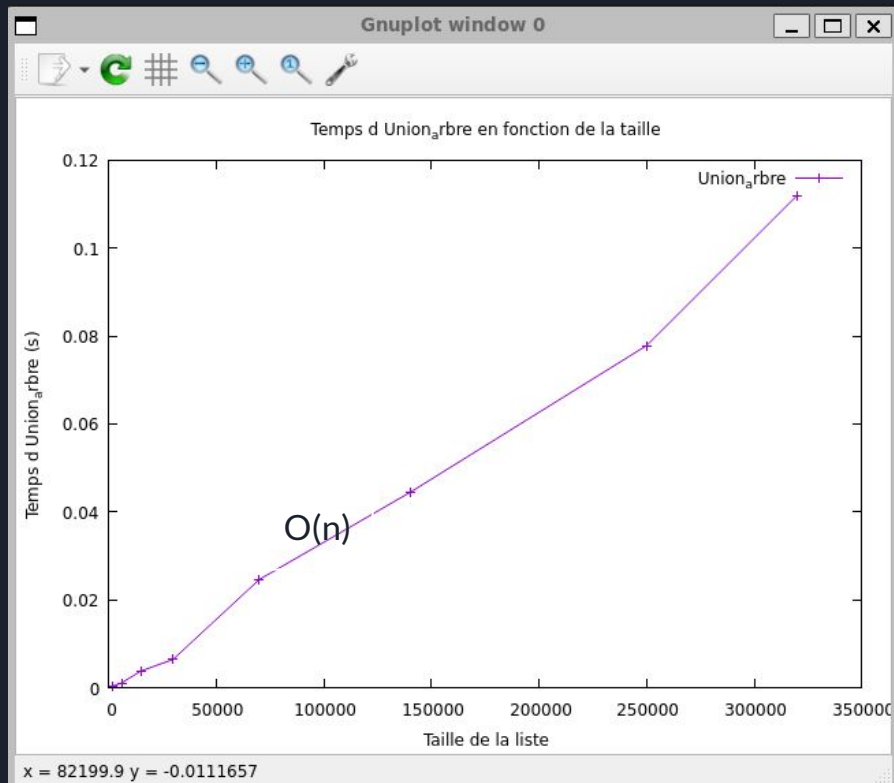
TAS PRIORITÉ MIN

VIA UN ARBRE BINAIRE

FONCTION : AJOUT ITÉRATIF ET CONSTRUCTION

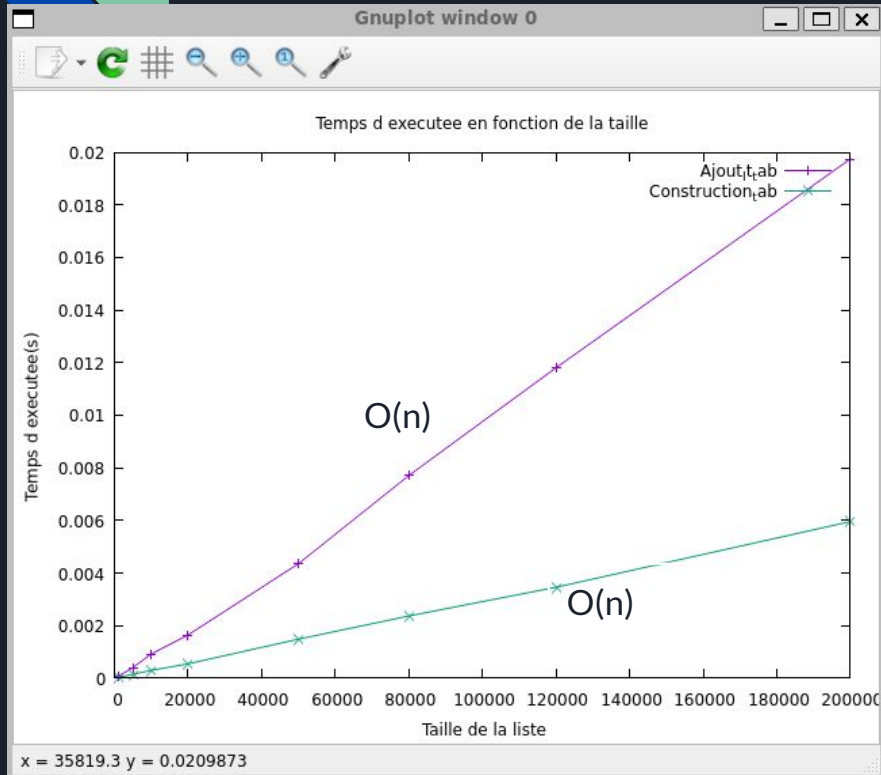


FONCTION : UNION

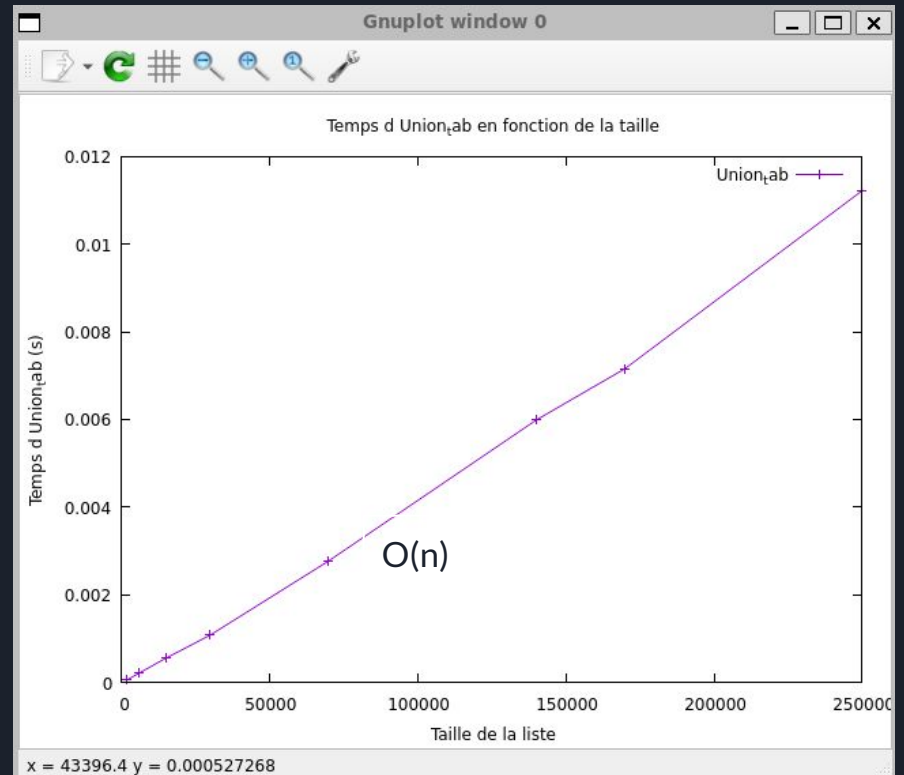


VIA UN TABLEAU

FONCTION : AJOUT ITÉRATIF ET CONSTRUCTION



FONCTION : UNION





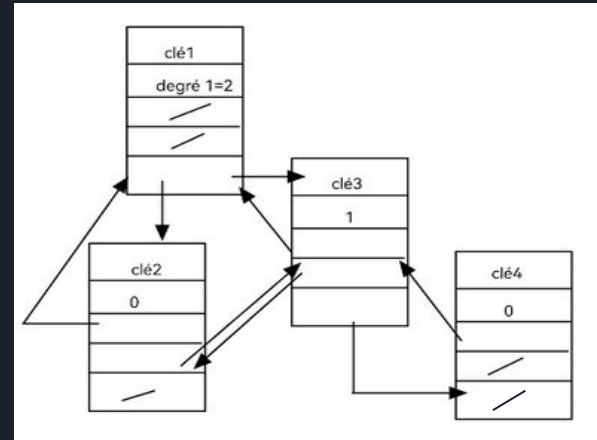
PARTIE 3

LES FILES BINOMIALES

STRUCTURES DE DONNÉES

TOURNOI BINOMIALE & FILE BINOMIALE

```
//Tournois Binomiale Noued
typedef struct tournoisB{
    Cle_entier* cle; // Valeur s
    int degre; // Degré du nœud
    struct tournoisB* parent; //
    struct tournoisB* enfant; //
    struct tournoisB* frere; //
} TournoisB;
```



exemple darbre binomial

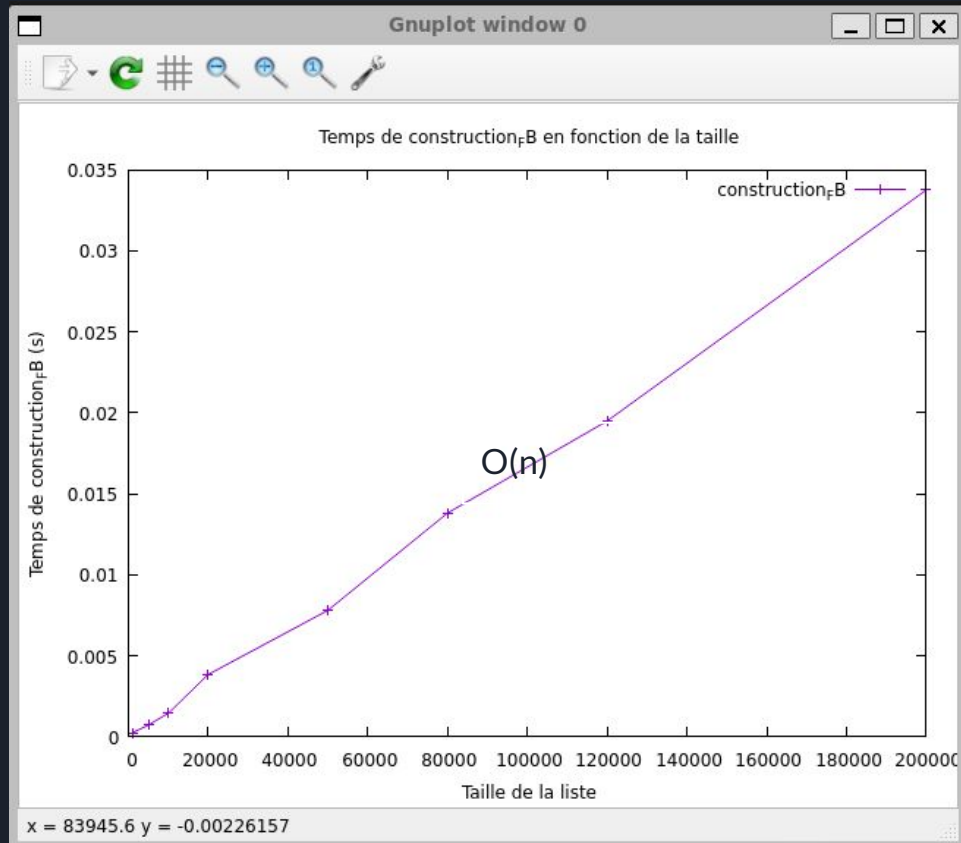
```
//File Binomiale
typedef struct tb_cell{
    TournoisB* tb;
    struct tb_cell* suiv;
}TB_Cell;
```



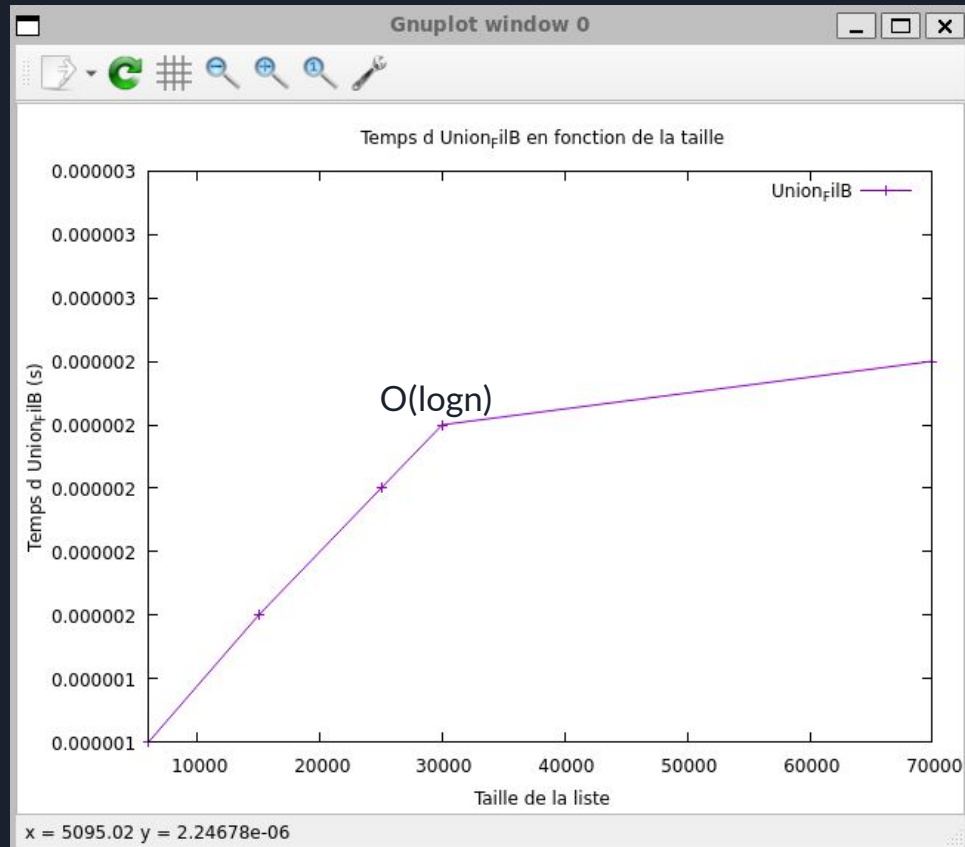
COMPLEXITÉS TEMPORELLES

FILE BINOMIALE

FONCTION : CONSTRUCTION



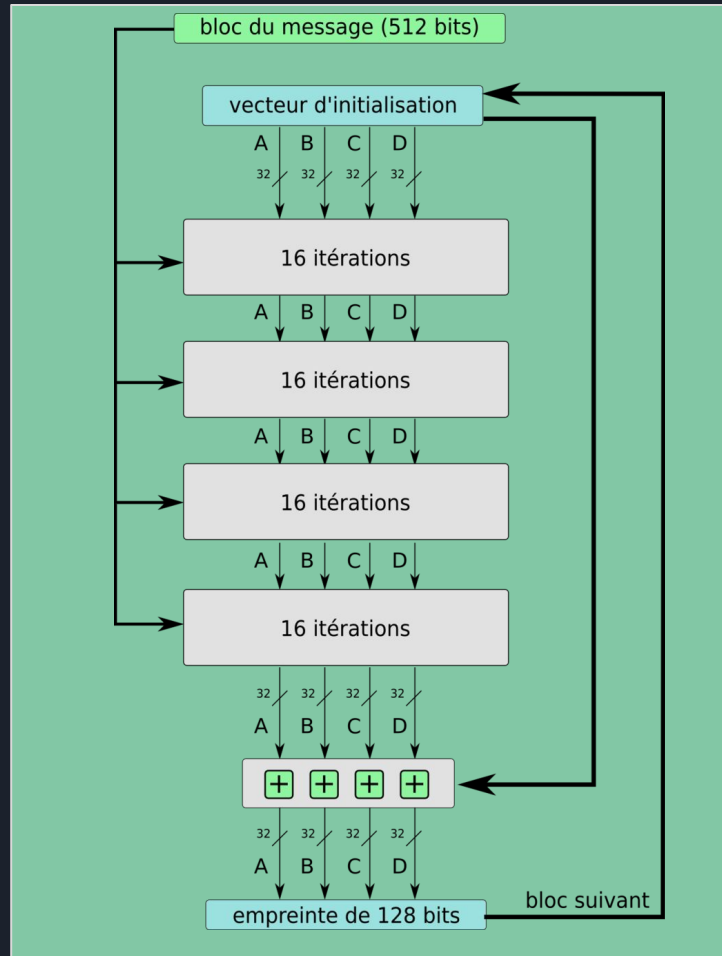
FONCTION : UNION





PARTIE 4

FONCTION DE HACHAGE





PARTIE 5

LES Arbres DE RECHERCHE



STRUCTURE DE DONNÉES

ARBRE DE RECHERCHE

```
typedef struct ab_rech{  
    Cle_entier* cle;  
    struct ab_rech* fils_g;  
    struct ab_rech* fils_d;  
    struct ab_rech* parent;  
}AB_Rech;
```



LES PRIMITIVES

- RACINE
- SOUS_ARBRE_GAUCHE
- SOUS_ARBRE_DROIT
- PARENT
- ABR_AJOUT

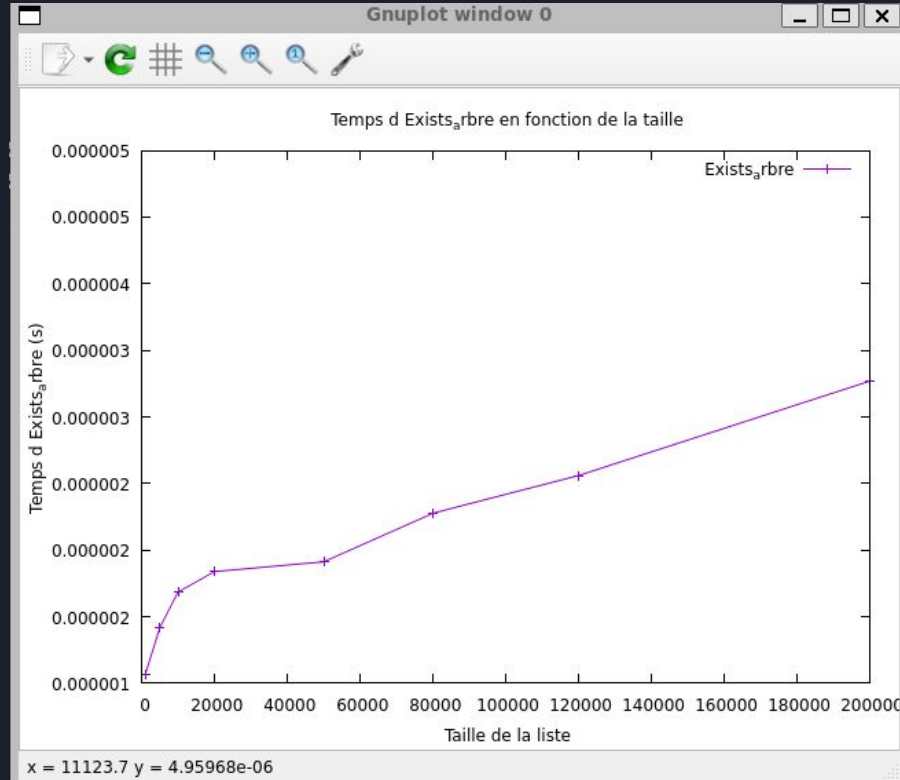


COMPLEXITÉS TEMPORELLES

ARBRES BINAIRES DE RECHERCHE

ARBRES BINAIRE DE RECHERCHE

FONCTION : EXISTS





PARTIE 6

ETUDE EXPÉRIMENTALE



STRUCTURES DE DONNÉES

STRUCTURES RÉELLES : LES MOTS

UNE LISTE DE MOTS

```
typedef struct words{  
    char* data;  
    struct words* suiv;  
}Words;
```

DESCRIPTION

Fichier “Shakespeare” composé de 23086 mots distincts

Il n’y a pas de collisions détectées car les empreintes des clés sont toutes distincts deux-à-deux (valeurs de hachage toutes distincts)

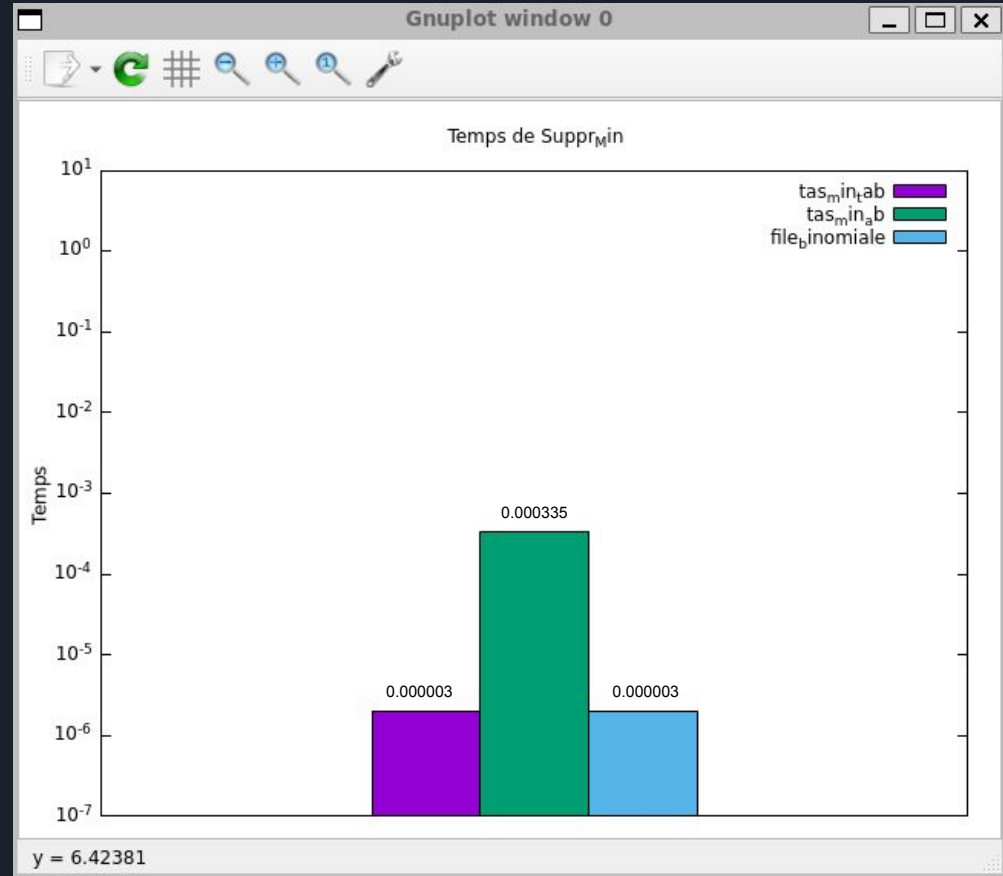


COMPARAISON GRAPHIQUE DES TEMPS D'EXÉCUTION

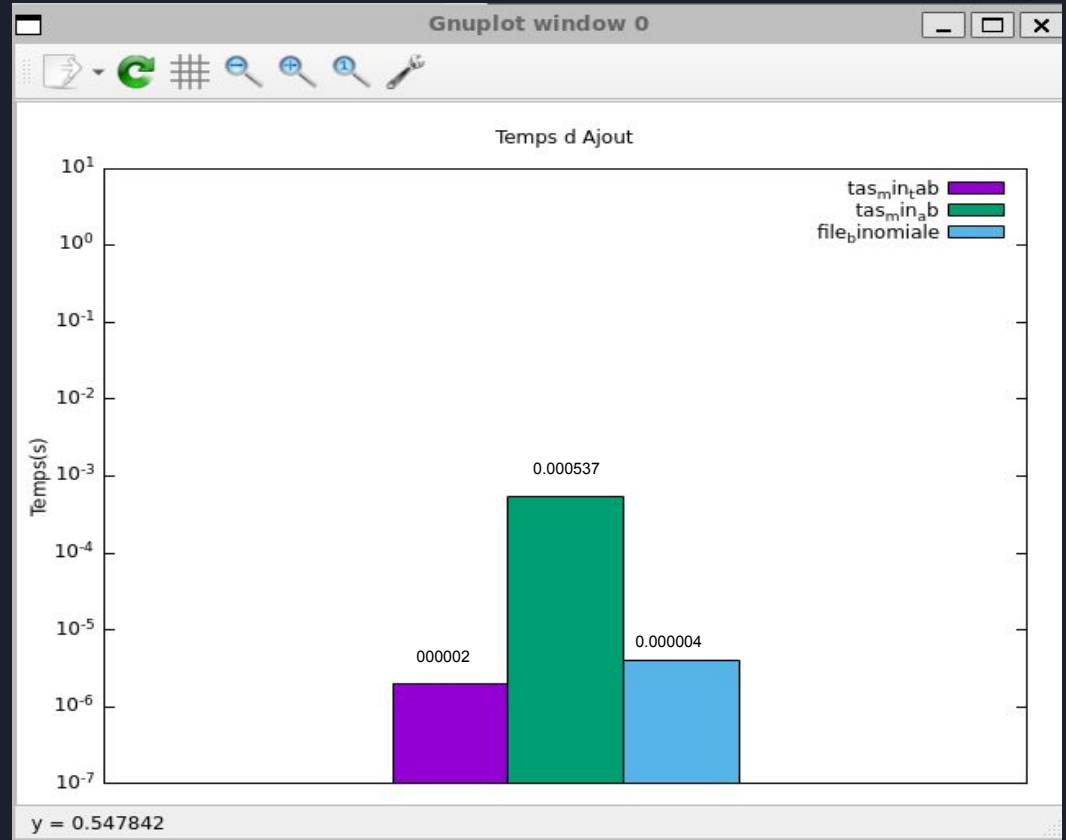
TAS MIN & FILE BINOMIALE

TAS MIN & FILE BINOMIALE

FONCTION : SUPPR MIN

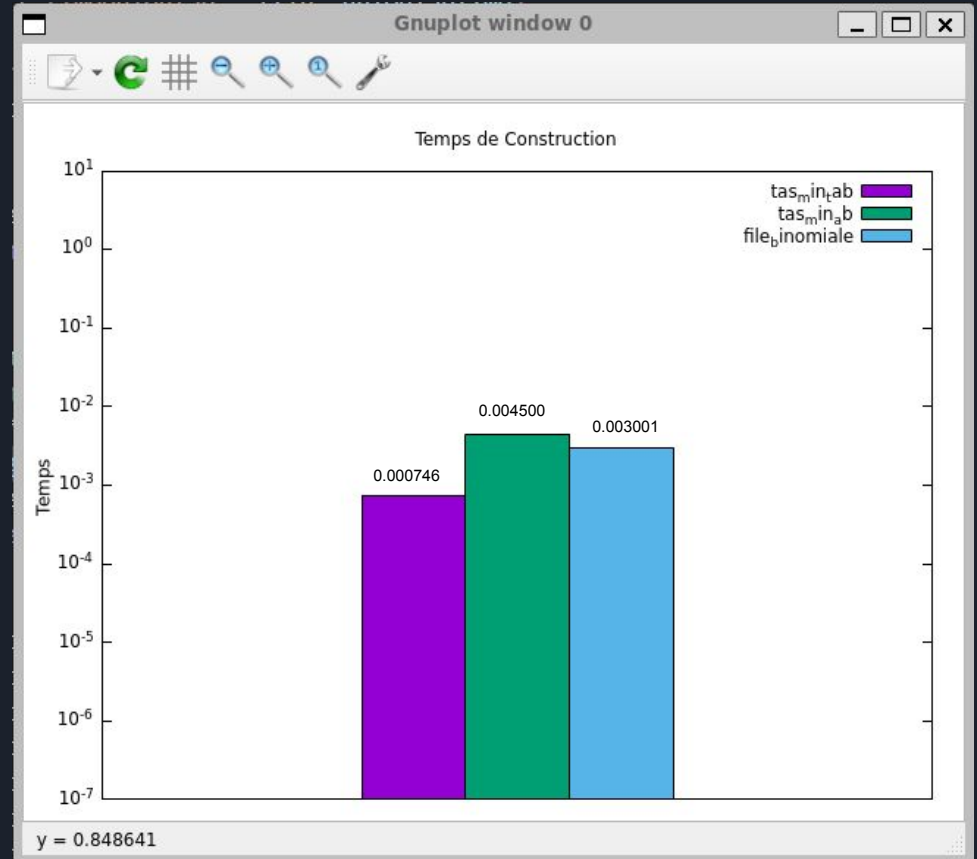


FONCTION : AJOUT



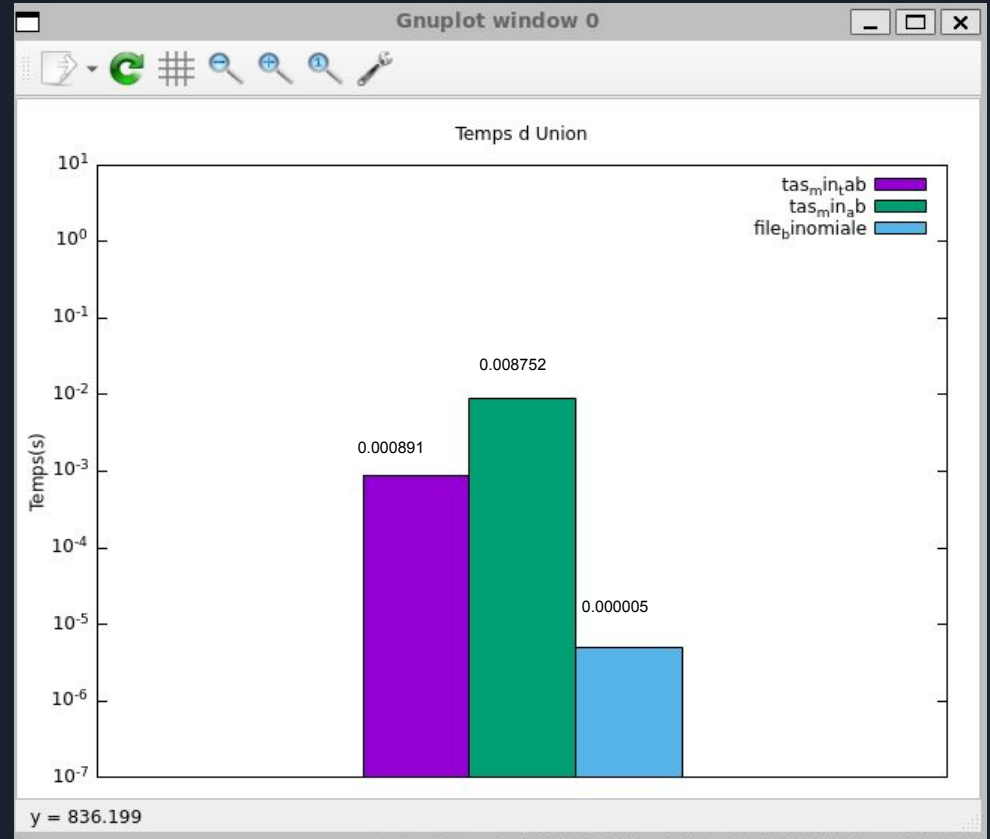
TAS MIN & FILE BINOMIALE

FONCTION : CONSTRUCTION



TAS MIN & FILE BINOMIALE

FONCTION : UNION





**MERCI POUR VOTRE
ATTENTION**