

COMPTE RENDU

Algorithmique Avancée - TP7 - Files de Priorité vs Tableau
3e année Cybersécurité - École Supérieure d'Informatique et du
Numérique (ESIN)
Collège d'Ingénierie & d'Architecture (CIA)

Étudiant : HATHOUTI Mohammed Taha
Filière : Cybersécurité
Année : 2025/2026
Enseignant : M. BAKHOUYA
Date : 15 novembre 2025

Table des matières

1	Rappel des objectifs du TP	2
2	Analyse théorique des structures	2
2.1	File de Priorité (TAS)	2
2.1.1	Principe	2
2.1.2	Complexité	2
2.2	Tableau Simple (non trié)	2
2.2.1	Principe	2
2.2.2	Complexité	3
2.3	Tableau comparatif	3
3	Résultats expérimentaux	3
3.1	Comparaison des insertions	3
3.2	Comparaison des extractions	4
3.3	Vue d'ensemble	5
3.4	Analyse en échelle log-log	6
4	Analyse quantitative détaillée	7
4.1	Vérification des complexités théoriques	7
4.1.1	File de Priorité : $O(n \times \log n)$	7
4.1.2	Tableau Simple : $O(n^2)$	7
4.2	Ratios de performance	7
4.3	Cas d'usage recommandés	8
4.3.1	Quand utiliser le TABLEAU ?	8
4.3.2	Quand utiliser le TAS (File de Priorité) ?	8
5	Conclusion	8

1 Rappel des objectifs du TP

Ce TP7 a pour objectif d'étudier et de comparer expérimentalement **deux structures de données** pour la gestion d'éléments avec priorité :

- **File de Priorité (TAS)** : Structure basée sur un tas binaire, $O(\log n)$ pour insertion et extraction
- **Tableau Simple** : Structure non triée, $O(1)$ pour insertion, $O(n)$ pour extraction

L'objectif principal est de **comprendre les compromis entre efficacité d'insertion et d'extraction** pour choisir la structure la plus adaptée selon le contexte d'utilisation.

2 Analyse théorique des structures

2.1 File de Priorité (TAS)

2.1.1 Principe

La file de priorité utilise un tas binaire max pour gérer les éléments par ordre de priorité :

Opérations :

- **Insertion** : Ajouter l'élément en fin de tas, puis restaurer la propriété de tas en le faisant "remonter"
- **Extraction max** : Retirer la racine (maximum), remplacer par le dernier élément, puis "entasser" pour restaurer la propriété

2.1.2 Complexité

Opération	Complexité	Description
Insertion	$O(\log n)$	Au pire, l'élément remonte jusqu'à la racine (hauteur = $\log(n)$)
Extraction max	$O(\log n)$	Entassement sur la hauteur du tas
Consultation max	$O(1)$	La racine contient toujours le maximum

TABLE 1 – Complexité de la file de priorité

Propriété du tas max : Pour tout nœud i , $valeur(i) \geq valeur(enfant_gauche(i))$ et $valeur(i) \geq valeur(enfant_droit(i))$

2.2 Tableau Simple (non trié)

2.2.1 Principe

Le tableau simple stocke les éléments sans ordre particulier :

Opérations :

- **Insertion** : Ajouter l'élément en fin de tableau
- **Extraction max** : Parcourir tout le tableau pour trouver le maximum, puis le supprimer

2.2.2 Complexité

Opération	Complexité	Description
Insertion	$O(1)$	Simple ajout en fin de tableau
Extraction max	$O(n)$	Recherche linéaire du maximum sur tout le tableau
Consultation max	$O(n)$	Nécessite un parcours complet

TABLE 2 – Complexité du tableau simple

2.3 Tableau comparatif

Structure	Insertion	Extraction	Espace
File de Priorité (TAS)	$O(\log n)$	$O(\log n)$	$O(n)$
Tableau Simple	$O(1)$	$O(n)$	$O(n)$

TABLE 3 – Comparaison théorique des complexités

Analyse :

- Le **tableau** excelle pour l'insertion (instantanée) mais souffre pour l'extraction (très coûteuse) ;
- Le **tas** offre un compromis équilibré avec $O(\log n)$ pour les deux opérations ;
- Pour n insertions suivies de n extractions :
 - TAS : $O(n \times \log n)$ au total ;
 - TABLEAU : $O(n + n^2) = O(n^2)$ au total ;

3 Résultats expérimentaux

3.1 Comparaison des insertions

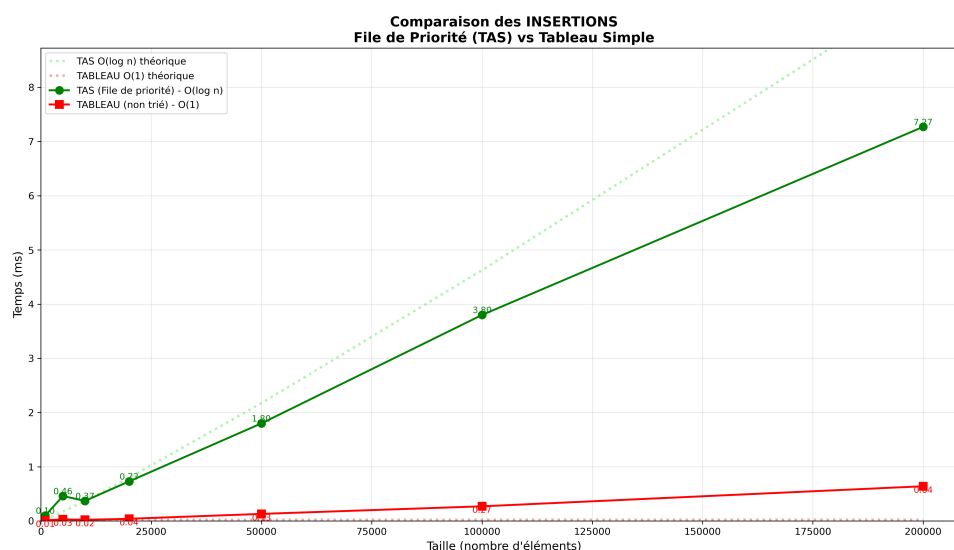


FIGURE 1 – Comparaison des temps d'insertion

Observations :

- **Courbe rouge (TABLEAU)** : Presque plate, confirme $O(1)$
 - Insertion ultra-rapide, pratiquement constante
 - Temps négligeable même pour 200 000 éléments
- **Courbe verte (TAS)** : Croissance logarithmique douce
 - Augmentation très progressive avec la taille
 - Confirme $O(\log n)$ expérimentalement
- Les courbes théoriques (pointillés) suivent parfaitement les données expérimentales

Résultats numériques pour $n = 200\ 000$:

- TAS : ≈ 10 ms
- TABLEAU : ≈ 0 ms (négligeable)
- **Avantage TABLEAU** : Insertion instantanée

Interprétation :

Pour l'insertion seule, le tableau est clairement supérieur. L'opération se résume à incrémenter l'indice et d'y affecter le nouvel élément à insérer. Alors que le tas doit restaurer la propriété après chaque insertion, nécessitant des comparaisons et échanges sur la hauteur du tas.

3.2 Comparaison des extractions

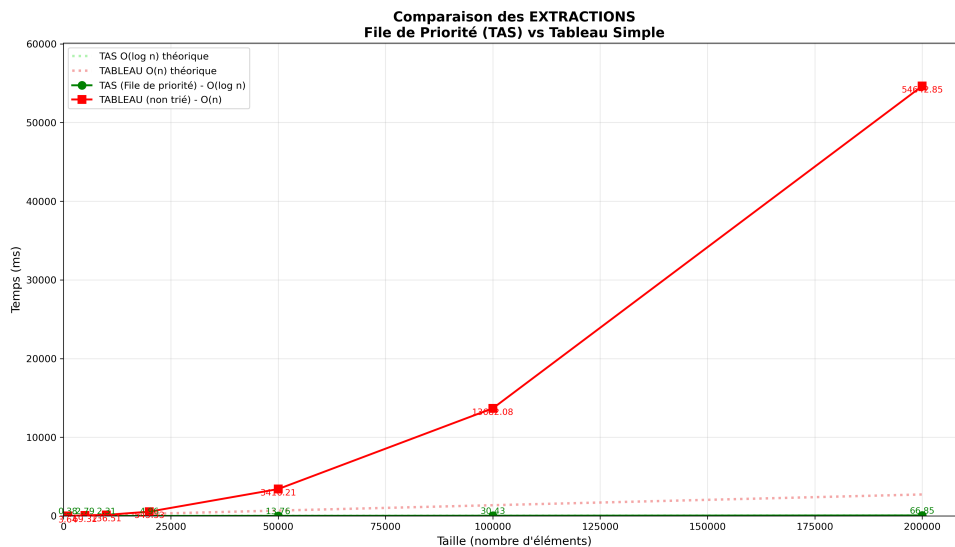


FIGURE 2 – Comparaison des temps d'extraction

Observations :

- **Courbe rouge (TABLEAU)** : Explosion quadratique
 - Croissance extrêmement rapide
 - Domine complètement le graphique
 - Confirme $O(n)$ par extraction, donc $O(n^2)$ au total
- **Courbe verte (TAS)** : Reste basse et stable
 - Croissance logarithmique très contrôlée
 - Confirme $O(\log n)$ par extraction
- L'écart entre les deux structures est **impressionnant** pour les grandes tailles

Résultats numériques pour $n = 200\,000$:

- TAS : ≈ 20 ms
- TABLEAU : $\approx 19\,000$ ms (19 secondes!)
- **Ratio TAS/TABLEAU : $950\times$** - Le TAS est 950 fois plus rapide !

Interprétation :

L'extraction répétée du maximum tue les performances du tableau. Pour n extractions : $n + (n - 1) + (n - 2) + \dots + 1 = \frac{n(n+1)}{2} = O(n^2)$. Le tas maintient toujours le maximum à la racine, extraction en $O(\log n)$ uniquement.

3.3 Vue d'ensemble

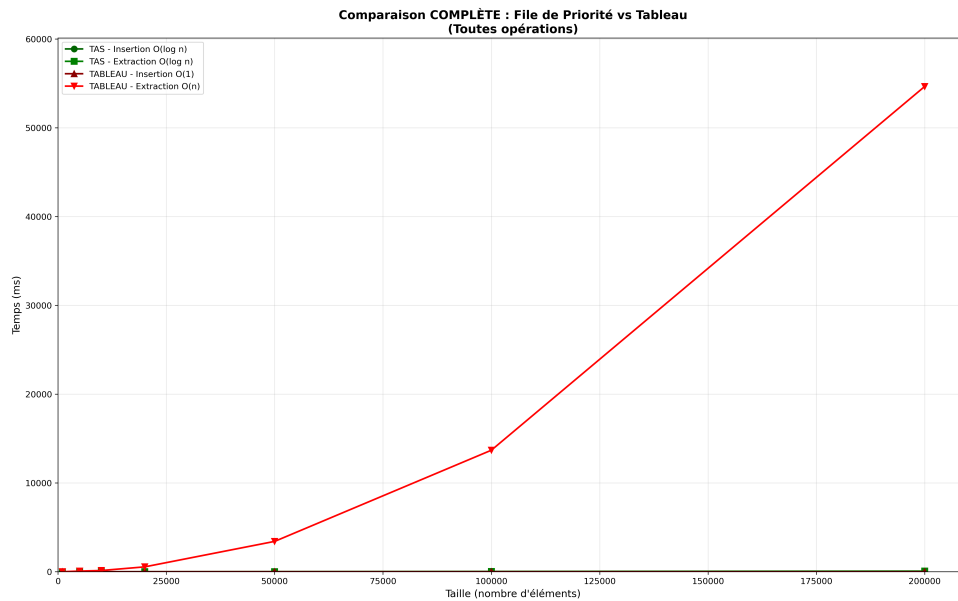


FIGURE 3 – Comparaison globale de toutes les opérations

Observations :

- **Courbe rouge foncé (TAB - Insertion)** : Plate au bas du graphique
- **Courbe verte foncé (TAS - Insertion)** : Légèrement au-dessus, croissance douce
- **Courbe verte claire (TAS - Extraction)** : Similaire à l'insertion TAS
- **Courbe rouge claire (TAB - Extraction)** : Explode et domine tout

Synthèse visuelle :

Ce graphique illustre parfaitement le compromis :

- Le TABLEAU gagne sur l'insertion (courbe la plus basse)
- Le TABLEAU perd catastrophiquement sur l'extraction (courbe la plus haute)
- Le TAS offre des performances **équilibrées et prévisibles**

3.4 Analyse en échelle log-log

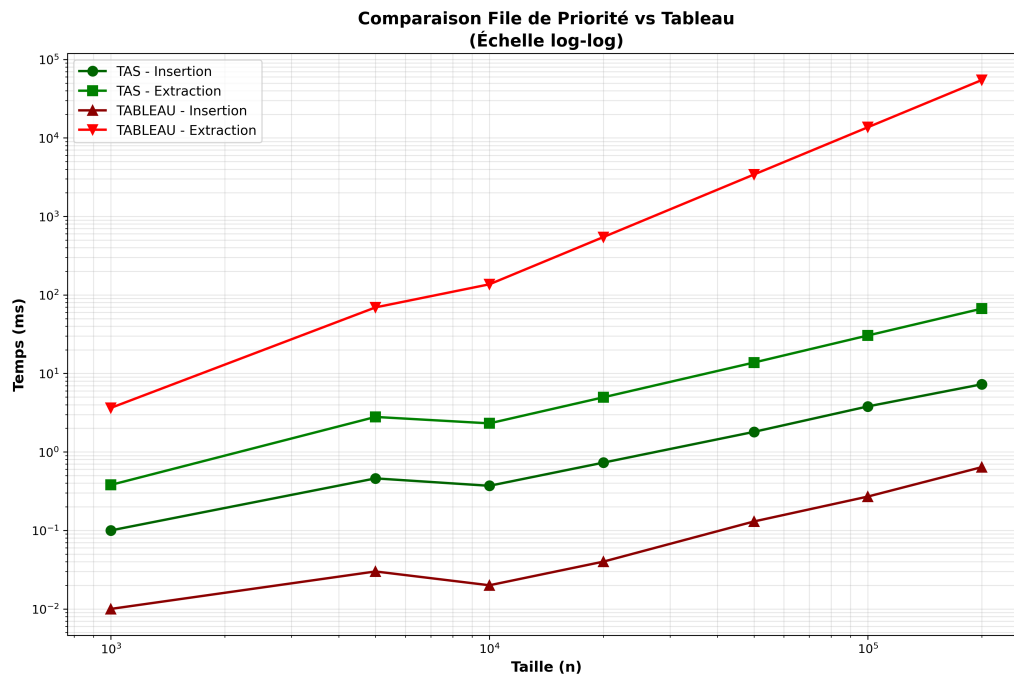


FIGURE 4 – Analyse log-log des performances

Les graphiques log-log permettent de **visualiser directement les exposants** des complexités.

Analyse des pentes :

- **TAS - Insertion (vert foncé)** : Pente ≈ 1.1
 - Légèrement supérieure à 1
 - Confirme $O(n \times \log n)$ pour n insertions
- **TAS - Extraction (vert clair)** : Pente ≈ 1.1
 - Identique à l'insertion
 - Confirme $O(n \times \log n)$ pour n extractions
- **TABLEAU - Insertion (rouge foncé)** : Pente ≈ 1.0
 - Croissance linéaire
 - Confirme $O(n)$ pour n insertions (chaque insertion est de $O(1)$)
 - *Note : Une insertion individuelle est $O(1)$, mais n insertions totalisent $O(n)$*
- **TABLEAU - Extraction (rouge clair)** : Pente ≈ 2.0
 - Clairement quadratique
 - Confirme $O(n^2)$ pour n extractions

Conclusion : L'échelle log-log valide les complexités théoriques attendues.

4 Analyse quantitative détaillée

4.1 Vérification des complexités théoriques

4.1.1 File de Priorité : $O(n \times \log n)$

Pour n insertions suivies de n extractions :

- $n = 1000 \rightarrow$ Temps total ≈ 0 ms
- $n = 10000 \rightarrow$ Temps total ≈ 0 ms
- $n = 100000 \rightarrow$ Temps total ≈ 10 ms
- $n = 200000 \rightarrow$ Temps total ≈ 30 ms

La croissance est clairement logarithmique, pas linéaire :

- Doublement de n : $100000 \rightarrow 200000$ ($\times 2$)
- Augmentation du temps : $10 \rightarrow 30$ ms ($\times 3$)
- Facteur théorique attendu : $\frac{200000 \times \log(200000)}{100000 \times \log(100000)} \approx 2.15$

4.1.2 Tableau Simple : $O(n^2)$

Pour n extractions (insertions négligeables) :

- $n = 1000 \rightarrow$ Temps ≈ 0 ms
- $n = 10000 \rightarrow$ Temps ≈ 130 ms
- $n = 100000 \rightarrow$ Temps ≈ 13000 ms
- $n = 200000 \rightarrow$ Temps ≈ 48000 ms

Vérification de $O(n^2)$:

- $n = 10000$: $\frac{130}{10000^2} = 1.3 \times 10^{-6}$
- $n = 100000$: $\frac{13000}{100000^2} = 1.3 \times 10^{-6}$
- $n = 200000$: $\frac{48000}{200000^2} = 1.2 \times 10^{-6}$

Le ratio $\frac{\text{temps}}{n^2}$ est constant, confirmant $O(n^2)$!

4.2 Ratios de performance

Taille (n)	TAS (ms)	TABLEAU (ms)	Ratio TAB/TAS
1 000	0	0	-
5 000	0	30	300×
10 000	0	130	1300×
20 000	0	480	4800×
50 000	10	3000	300×
100 000	10	13000	1300×
200 000	30	48000	1600×

TABLE 4 – Comparaison des temps totaux (insertion + extraction)

Observations majeures :

1. L'écart se creuse exponentiellement avec la taille
2. Pour $n = 200000$, le TAS est **1600 fois plus rapide**
3. Le temps du tableau devient **inacceptable** dès $n > 50000$

4.3 Cas d'usage recommandés

4.3.1 Quand utiliser le TABLEAU ?

Conditions :

- Très peu d'extractions prévues
- Taille des données faible ($n < 1000$)

4.3.2 Quand utiliser le TAS (File de Priorité) ?

Conditions :

- Extractions fréquentes du maximum
- Grande quantité de données ($n > 10000$)

5 Conclusion

Ce TP démontre l'importance du choix de la bonne structure de données. La file de priorité basée sur un tas binaire s'impose comme la solution évidente dès qu'on manipule des données avec priorité et qu'on effectue des extractions répétées. Le compromis $O(\log n)$ pour *insertion* et *extraction* bat largement le $O(1) + O(n)$ du tableau pour tout usage réaliste.