

---

## Files d'attente à priorités

---

### 1) Explication du principe des algorithmes utilisés pour l'implémentation des primitives :

Deux structures :

- Une structure *Elément* contenant deux attributs (un entier *donnée* et un entier *priorité*) ;
- Un deuxième structure *File* contenant deux attributs (un tableau d'*Elément* *tab* et un entier *taille*).

Cinq fonctions :

- **Initialiser** allouant l'espace nécessaire (taille maximum x taille d'un *Elément*) au tableau *tab* et initialisant la *taille* à zéro ;
- **EstVide** qui teste si la file est vide en comparant la valeur de *taille* avec zéro ;
- **EstPleine** qui teste si la file est pleine en comparant la valeur de *taille* avec zéro ;
- **Inserer** qui ajoute un *Elément* a l'indice *taille* (éléments non triés) et incrémente la valeur de *taille* ;
- **Extraire** qui recherche l'*Elément* possédant la plus grande *priorité* et le retourne avec le pointeur en paramètre puis place le dernier *Elément* à la place de celui-ci afin que les *Eléments* restent contiguës.

La taille maximale est une constante définie dans le fichier file.h.

### 2) Présentation commentée des tests que vous avez effectués et leurs objectifs :

Tableau des tests avec des *Eléments* ajoutés dans l'ordre croissant :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	1	2	3	4	4	3	2	1
Priorité	1	2	3	4	4	3	2	1

Tableau des tests avec des *Eléments* ajoutés dans l'ordre décroissant :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	4	3	2	1	4	3	2	1
Priorité	4	3	2	1	4	3	2	1

Tableau des tests avec des Eléments ajoutés dans l'ordre aléatoire :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	2	4	1	3	4	3	2	1
Priorité	2	4	1	3	4	3	2	1

Tableau des tests avec des couples aléatoires ajoutés dans l'ordre aléatoire :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	2	1	2	3	2	1	2	3
Priorité	4	3	2	1	4	3	2	1

Tableau des tests avec en alternant insertion / extraction d'Eléments :

Type d'évènement	<i>i</i>	<i>i</i>	<i>e</i>	<i>i</i>	<i>e</i>	<i>i</i>	<i>e</i>	<i>e</i>
Donnée	1	2	2	4	4	3	3	1
Priorité	1	2	2	4	4	3	3	1

Tableau des tests avec une majorité d'extraction que d'insertion :

Type d'évènement	<i>i</i>	<i>i</i>	<i>e</i>	<i>i</i>	<i>e</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	1	2	2	4	4	3	3	1	File vide
Priorité	1	2	2	4	4	3	3	1	

Tableau des tests en dépassant la taille maximale :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>
Donnée	1	2	2	4	4	3	3	1	File pleine
Priorité	1	2	2	4	4	3	3	1	

Tableau des tests avec des priorités identiques :

Type d'évènement	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
Donnée	1	2	2	4	2	2	4	1
Priorité	1	2	3	2	3	2	2	1

### 3) Présentation de l'algorithme utilisé pour le programme de simulation :

Non réalisé.