

# Structures de données

## TD - 1

1. Bibliothèque pour manipuler les matrices (à deux dimensions).
  - (a) Écrivez la structure de donnée pour stocker une matrice de N lignes et M colonnes.
  - (b) Écrivez la fonction pour allouer une matrice de N lignes et M colonnes.
  - (c) Écrivez la fonction pour libérer une matrice.
  - (d) Écrivez les fonctions pour lire et écrire dans la matrice.
  - (e) Écrivez la fonction pour calculer la somme des éléments d'une matrice.
  - (f) Écrivez la fonction pour calculer le produit matriciel entre deux matrices. Pour rappel, la multiplication entre deux matrices  $\mathbf{A} \in \mathcal{M}_{M,N}$  et  $\mathbf{B} \in \mathcal{M}_{N,P}$  s'écrit :

$$c_{i,j} = \sum_{k=1}^N a_{i,k} \times b_{k,j} \quad (1)$$

avec  $c_{i,j}$  la  $i$ -ème ligne et  $j$ -ème colonne de la matrice  $\mathbf{C} \in \mathcal{M}_{M,P}$ .

2. File (FIFO) à buffer circulaire

Un buffer circulaire (Figure 1) est une structure de données utilisant un buffer de taille fixe et dont le début et la fin sont considérés comme connectés.

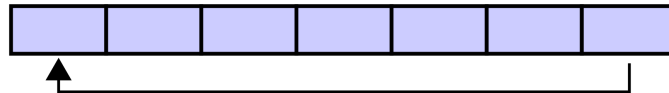


FIGURE 1 – Buffer circulaire vide

- (a) Écrivez la structure de donnée pour gérer une file (FIFO) dans un buffer circulaire.
  - (b) Écrivez la fonction d'insertion d'un éléments dans la file.
  - (c) Écrivez la fonction de suppression d'un éléments dans la file.
3. Recherche des k plus petits éléments. Nous considérons T un tableau non trié de n entiers.
  - (a) Écrivez un algorithme qui calcule les k plus petits éléments en procédant par recherches successives.
  - (b) Écrivez un algorithme qui calcule les k plus petits éléments en triant le tableau au préalable (la fonction de tri par ordre croissant est donnée  $\text{tri}(T)$  ; son coût est :  $n \log(n)$ )
  - (c) Comparez le temps mis par chacun des algorithmes pour  $k = 1..n$ .

4. Compléter le tableau en indiquant la complexité dans le pire cas pour chaque opération et chaque structure de donnée. L'ensemble des données noté  $S$  admet un ordre total, et  $x$  désigne un élément.

	liste non triée simplement chaînée	liste triée simplement chaînée	liste non triée doublement chaînée	liste triée doublement chaînée	tableau dynamique non triée	tableau dynamique triée
Rechercher( $x, S$ )						
Insérer( $x, S$ )						
Supprimer( $x, S$ )						
ValeurSuivante( $x, S$ )						
ValeurPrécédente( $x, S$ )						
Minimum( $S$ )						
Maximum( $S$ )						

Remarques : Rechercher( $x, S$ ) renvoie la position de  $x$  dans  $S$  s'il est présent (et sinon NULL), les opérations Supprimer( $x, S$ ), ValeurSuivante( $x, S$ ), ValeurPrécédente( $x, S$ ) supposent que l'on connaît déjà la position de  $x$  dans  $S$  (appel à Rechercher( $x, S$ ) déjà fait).