

Élément de module : Complexité algorithmique et structure de données

Examen de rattrapage durée : 90 minutes documents non autorisés

On peut utiliser et appeler les fonctions ou algorithmes classiques de manipulation de structures de données sans les réimplémenter

Exercice 1 : Piles et files (10 points)

Nous disposons d'un tableau T de n files $F_1, F_2, \dots, F_n$. Chaque file i contient n_i valeurs entières triées selon l'ordre croissant.

1. Écrire un algorithme ou une fonction **fusionner** qui fusionne l'ensemble d'éléments de toutes les files du tableau T dans une seule file dont les éléments sont également triés de la même manière
2. Calculez la complexité temporelle de cet algorithme en fonction des tailles des files fusionnées

Exercice 2 : Arbre binaires (10 points)

1. Insérer successivement les éléments suivants dans un arbre binaire de recherche : 10, 30, 11, 12, 4, 6, 8, 9, 15. (Donner l'arbre après chaque insertion)
2. Supprimer successivement les éléments 10, 8, 11 de l'arbre précédent en montrant l'arbre résultat après chaque suppression
3. Écrire un algorithme qui prend en paramètre un arbre binaire et retourne la somme des éléments de ses nœuds doubles.
4. Écrire un algorithme qui prend en paramètre arbre binaire de recherche en affiche ces éléments selon l'ordre décroissant.