

Exercice 1 : Analyse et Implémentation

Écris une fonction qui recherche une valeur dans un tableau trié en utilisant la **recherche dichotomique** (recherche binaire). Calcule et commente sa **complexité**.

Exercice 2 : Création d'un Programme de Tri Hybride

Implémente un programme qui :

1. Trie un tableau en utilisant **le tri à bulle** si la taille est inférieure à 10.
2. Passe au **tri rapide** pour des tailles supérieures.

Ajoute une option pour afficher le **temps d'exécution** de chaque méthode.

Exercice 3 : Parcours et Opérations sur Arbres

Écris un programme pour :

1. Construire un **arbre binaire** à partir d'un tableau donné (les éléments du tableau doivent être insérés séquentiellement).
 2. Effectuer les parcours suivants : **préfixé, infixé, postfixé**.
 3. Ajouter une méthode pour trouver **le plus grand élément** de l'arbre.
-

Exercice 4 : Mélange Récursivité et Arbres

Programme une fonction récursive qui :

1. Calcule la **profondeur** d'un arbre binaire.
2. Compte le nombre total de **nœuds feuilles**.

Exemple de sortie :

Profondeur de l'arbre : 3
Nombre de feuilles : 4

Exercice 5 : Tri avec Notion de Complexité

Écris un programme en C qui :

1. Trie un tableau en utilisant **le tri par insertion**.
2. Compte et affiche le **nombre total de comparaisons effectuées**.

3. Compare les performances avec **le tri par fusion** pour le même tableau.
-

Exercice 6 : Algorithme de Recherche dans Arbre

Programme une fonction récursive qui :

1. Recherche une valeur dans un **arbre binaire de recherche (ABR)**.
 2. Affiche le chemin suivi pour trouver la valeur (par exemple : racine -> gauche -> gauche).
-

Exercice 7 : Calcul sur Arbres avec Récursivité

Implémente un programme qui :

1. Ajoute tous les éléments d'un arbre binaire.
2. Affiche la somme totale et le nombre de nœuds parcourus.

Exemple de sortie :

Somme des nœuds : 45

Nombre de nœuds : 7

Exercice 8 : Tri et Recherches Multi-structures

Crée un programme qui :

1. Trie un tableau d'entiers aléatoires avec **le tri rapide**.
2. Insère les mêmes valeurs triées dans un **arbre binaire de recherche (ABR)**.
3. Recherche une valeur donnée dans le tableau (avec la recherche binaire) et dans l'arbre.

Compare les performances de la recherche dans les deux structures.

Exercice 9 : Mini-Système de Gestion

Programme un système de gestion de données simple en C :

1. Les données sont initialement dans un **tableau**.
 2. Le programme propose les options suivantes :
 - Trier les données (tri par sélection ou insertion).
 - Rechercher une valeur.
 - Ajouter les données dans un **arbre binaire**.
 - Afficher les parcours de l'arbre.
-

Exercice 10 : Simulation Algorithme de Compression

Simule un mini-algorithme qui :

1. Trie un tableau d'entiers par **tri fusion**.
 2. Stocke les entiers triés dans un **arbre binaire parfait**.
 3. Recherche les deux valeurs les plus petites pour simuler une compression (fusion de deux entiers).
-

Conseils

- **Difficulté croissante** : Commence par les exercices simples pour solidifier les bases avant d'essayer les plus complexes.
- **Structure des codes** : Utilise des fonctions pour chaque partie (ex : insertion dans l'arbre, tri, etc.) afin de garder le code clair.
- **Testez vos codes** avec différents ensembles de données pour être prêt à toute question piège.