

Université de Perpignan Via Domitia  
Février 2026

Travaux Dirigeés : Arbres et Algorithmes

Exercice 1:

Écrire un programme en C permettant d'afficher les noeuds d'un arbre général selon le parcours BFS. Écrire l'ensemble de noeuds selon le niveau. Trouver la hauteur de l'arbre. On utilise une file de liste chaînée. Ensuite on utilise un tableau circulaire.

Exécuter votre programme sur un arbre.

Exercice 2: Écrire une fonction récursive permettant de chercher un noeud dans un arbre binaire de recherche. Exécuter votre programme sur l'arbre déjà construit (TD1).

Exercice 3: Écrire une fonction non récursive permettant de chercher un noeud dans un arbre binaire de recherche. Exécuter votre programme sur l'arbre déjà construit (TD1).

Exercice 4: Écrire une fonction permettant de trouver le nombre de noeuds dans un arbre binaire. Exécuter votre programme sur l'arbre déjà construit (TD1).

Exercice 5: Écrire un programme en C permettant de trouver la hauteur d'un arbre binaire. Exécuter votre programme sur l'arbre déjà construit (TD1).

Exercice 5: Écrire un programme en C permettant de supprimer un noeud contenant un entier  $x$ . Exécuter votre programme sur l'arbre déjà construit (TD1).

Exercice 6: Écrire une fonction non-réursive permettant de trouver un entier maximum dans un tableau de  $n$  entiers.

Exercice 7: Écrire un programme en C afin de construire un tournoi. Un tournoi est un arbre binaire dans lequel chaque noeud intérieur contient un entier que est le maximum de ses deux enfants.

Exercice 8: Construire un tas avec 44 55 12 42 94 18 6 67.  
Effectuer le tri par tas en montrant les différentes étapes.

Exercice 9: Écrire une fonction non-réursive permettant d'afficher les noeuds d'un arbre binaire dans l'ordre préfixe, infixé, postfixe.

Exercice 10: Écrire une fonction en C pour la saisie d'un arbre général avec la représentation : firstChild et rightSibling. Imprimer l'arbre, trouver la hauteur.