

### 1. Classe Nœud

✍ Créer un module `abr_noeud` contenant la classe `Noeud` vu dans les activités précédentes.  
La classe `Noeud` doit contenir les **méthodes** nécessaires pour réaliser les programmes qui suivent.

### 2. Trie d'une liste

✍ Écrire un programme qui permet de trier une liste d'entiers ou de caractères non vide et d'afficher le résultat.  
Le programme utilise une structure d'arbre binaire de recherche pour trier la liste (faire appel au module `abr_noeud`).

✍ Modifier le programme afin qu'il renvoie la liste triée.

### 3. Maximum d'une liste

✍ Écrire un programme qui permet de d'afficher le maximum d'une liste d'entiers ou de caractères non vide.  
Le programme utilise une structure d'arbre binaire de recherche (faire appel au module `abr_noeud`).

### 4. Minimum d'une liste

✍ Écrire un programme qui permet de renvoyer le minimum d'une liste d'entiers ou de caractères non vide.  
Le programme utilise une structure d'arbre binaire de recherche (faire appel au module `abr_noeud`).

### 5. Recherche d'une valeur dans une liste

✍ Écrire un programme qui permet de savoir si un élément est présent dans une liste d'entiers ou de caractères non vide.  
Le programme utilise une structure d'arbre binaire de recherche (faire appel au module `abr_noeud`).

### 6. Affiche un chemin

✍ Écrire un programme qui permet d'afficher le chemin de puis la racine jusqu'au nœud donnée.  
Le programme utilise une structure d'arbre binaire de recherche (faire appel au module `abr_noeud`).

✍ Modifier le programme afin qu'il renvoie le chemin sous forme d'une liste.