Travaux Pratiques n°2

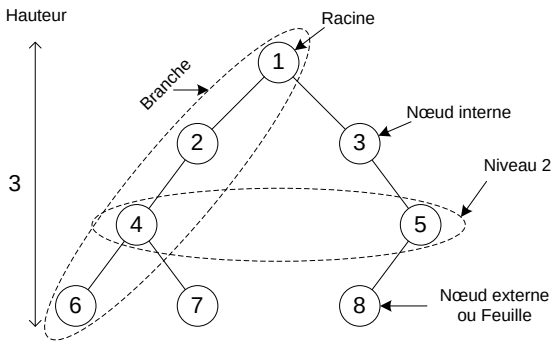
Cours d'Informatique de Deuxième Année

—Licence MI L2.2—

Arbres binaires

Jusqu'ici, nous avons vu les structures linaires comme les listes chaînées où un élément suit l'autre. Nous allons nous intéresser à des structures à deux dimensions appelées *arbres* qui sont au coeur d'un grand nombre d'algorithmes car ils permettent de hiérarchiser les données. Dans ce TP, nous allons nous intéresser à des arbres particuliers qu'on appelle les arbres binaires.

Dans notre vie quotidienne, nous rencontrons souvent la notion d'arbre. Par exemple, pour représenter l'organisation de compétitions sportives ou encore la hiérarchie d'une famille avec les arbres généalogiques. Une partie de la terminologie vient de cet usage.

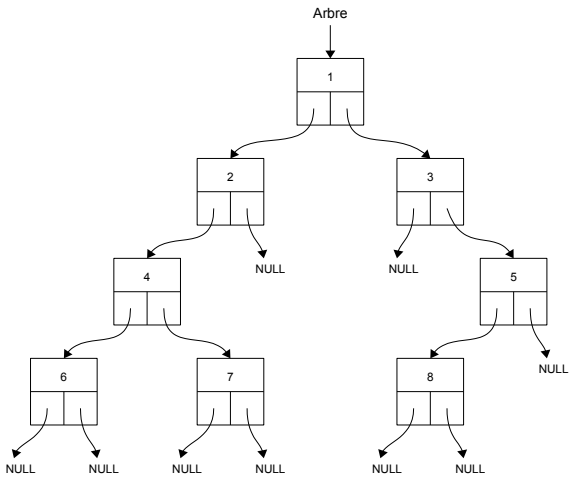


2, 3	enfant de 1
7	frère de 6
1, 2	ancêtre de 4
6, 7	descendant de 4

Un arbre binaire est un arbre dont chaque noeud possède au plus deux fils.

▸ **Exercice 1.** ***La structure d'arbre***

Définir un type `Arbre` *récursif contenant des entiers comme sur le schéma ci-dessous :*



▸ **Exercice 2.** ***Parcours en profondeur récursif***

Les arbres sont particulièrement adaptés à l'usage de la récursivité, écrire les fonctions suivantes en ***récursif*** *:*

- `int hauteurRec(Arbre a);` *calculant la hauteur d'un arbre binaire ;*
- `int nbNoeudsRec(Arbre a);` *calculant le nombre de noeuds d'un arbre binaire ;*
- `void affichagePrefixeRec(Arbre a);` *affichant un arbre en ordre préfixe, vous devez obtenir la sequence suivante : 1, 2, 4, 6, 7, 3, 5, 8 ;*
- `void affichageInfixeRec(Arbre a);` *affichant un arbre en ordre infixe, vous devez obtenir la sequence suivante : 6, 4, 7, 2, 1, 8, 5, 3 ;*
- `void affichageSuffixeRec(Arbre a);` *affichant un arbre en ordre suffixe, vous devez obtenir la sequence suivante : 6, 7, 4, 2, 8, 5, 3, 1.*

▸ **Exercice 3.** ***Parcours en profondeur itératif***

Le parcours en profondeur des arbres doit se faire récursivement, pour supprimer la récursivité, il faut utiliser une ***pile****. Récrire les fonctions précédentes en* ***itératif*** *à l'aide d'une pile.*

▸ **Exercice 4.** ***Parcours en largeur***

Nous voulons à présent parcourir l'arbre en ***largeur****, comment peut-on faire ce parcours ?*

Écrire une fonction affichant l'arbre en largeur. Vous devez obtenir la sequence suivante : 1, 2, 3, 4, 5, 6, 7, 8.