

## EA4 – Éléments d'algorithmique

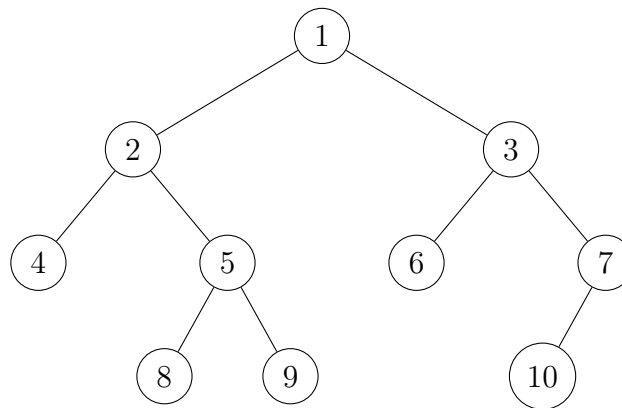
### TD n° 7

#### Exercice 1 :

1. Dénombrer les feuilles et les nœuds binaires de l'arbre binaire *parfait* de hauteur  $h$ .
2. Plus généralement, quel est le nombre de feuilles d'un arbre binaire *complet* à  $n$  nœuds binaires ? Et d'un arbre  $k$ -aire complet à  $n$  nœuds  $k$ -aires ?
3. Dessiner tous les arbres binaires complets à 1, 2 et 3 nœuds binaires.
4. Dessiner tous les arbres à 1, 2, 3 et 4 sommets.
5. Associer à chacun de ces arbres son image par la bijection « fils aîné – frère droit ».
6. Combien y a-t-il d'arbres binaires de recherche d'une forme donnée pour un ensemble de  $n$  valeurs fixées ?

#### Exercice 2 :

1. Lister les sommets de l'arbre binaire ci-dessous en ordre préfixe, infixé et suffixé.
2. Lister les sommets de l'arbre ci-dessous dans l'ordre d'un parcours en largeur d'abord (voir définition à l'exercice 6).



#### Exercice 3 :

Dessiner des arbres binaires de recherche de toutes les hauteurs possibles pour l'ensemble  $f1, 2, 3, 4, 5, 6, 7g$ .

#### Exercice 4 :

1. Dénombrer les arêtes d'un arbre quelconque à  $n$  sommets.
2. Quelle est la hauteur d'un arbre binaire parfait contenant  $2^k - 1$  sommets ?
3. Encadrer alors la hauteur  $h$  d'un arbre binaire en fonction du nombre  $n$  de ses sommets.

**Exercice 5 :**

1. Décrire un algorithme de calcul de la hauteur d'un arbre représenté par la liste chaînée des fils de chaque nœud. Quelle est sa complexité ?

On suppose maintenant que les arbres sont représentés par le tableau des pères : pour chaque  $i$ ,  $pere[i]$  est l'indice du père du sommet d'indice  $i$ . Par convention, la racine est son propre père.

2. Montrer que si la numérotation des sommets respecte la généalogie, *i.e.* si pour tout  $i$ ,  $pere[i] \leq i$ , alors la hauteur d'un arbre à  $n$  sommets peut encore être calculée en temps  $\mathcal{O}(n)$ .
3. (\*) Qu'en est-il si la numérotation est quelconque ?

**Exercice 6 :**

Un *parcours en largeur (d'abord)* d'un arbre visite ses nœuds niveau par niveau, par profondeur croissante : d'abord la racine, puis tous ses fils, puis tous les nœuds de profondeur 2, *etc.*

Donner un algorithme effectuant un parcours en largeur sur des arbres représentés par liste chaînée des fils. On pourra utiliser une structure de données supplémentaire pour stocker la liste des nœuds découverts mais pas encore visités.

**Exercice 7 :**

Donner un algorithme qui teste si un arbre est un ABR.