# DM arbres binaires



## Étude théorique

### Rappel

Soit $n$ un entier naturel, on a

$$\sum_{k=0}^{k=n} 2^k = 2^{n+1} - 1$$

C'est à dire que

$$1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$$

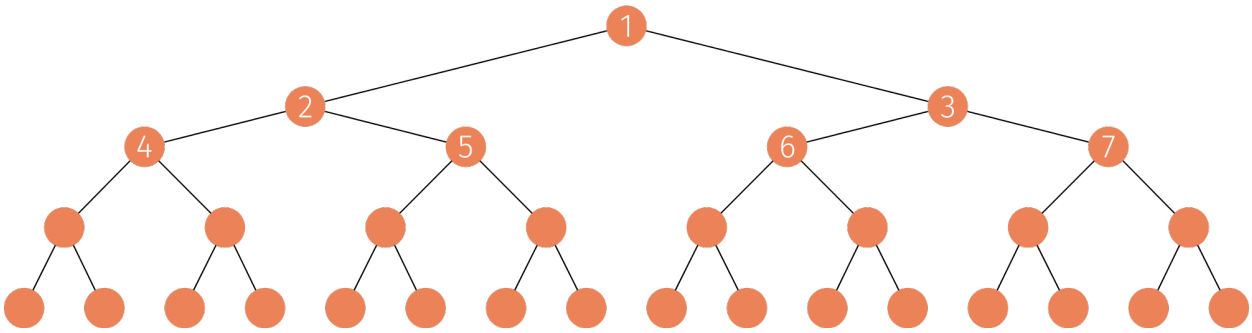
Ou encore

$$(\underbrace{1 \ldots 1}_{n \text{ bits}})_2 = 2^{n+1} - 1$$

On considère un arbre binaire **parfait** et on numérote ses nœuds de la manière suivante :

- la racine a le numéro 1 ;
- les 2 nœuds de hauteur 1 sont numérotés de gauche à droite respectivement 2 et 3 ;
- les 4 nœuds de hauteur 2 sont numérotés de gauche à droite respectivement 4, 5, 6 et 7 ;
- et ainsi de suite.

1. Compléter la numération de cet arbre de hauteur 4.




2. On considère un arbre binaire **parfait** de hauteur $h \in \mathbf{N}$.
   Soit $p$ un entier naturel inférieur ou égal à $h$, combien y a-t-il de nœuds de hauteur $p$ dans l'arbre ?
   Comment prouver ce résultat ?

3. On considère un nœud $\mathcal{N}$ de hauteur $p$ et on note $i(\mathcal{N})$ son numéro.
   Que vaut $i(\mathcal{N})$ au minimum ? Au maximum ?
   Comment prouver ces affirmations ? On pourra utiliser le rappel.

4. Quels sont les numéros du fils gauche de $\mathcal{N}$ et du fils droit de $\mathcal{N}$ ? Comment prouver ces résultats ?

5. Soit un nœud numéroté $k \geqslant 2$, quel est le numéro de son père ?

## Implémentation

On décide de représenter un arbre parfait de hauteur $h$ avec une liste de la bonne taille, l'élément d'indice 0 ne servant pas, celui d'indice 1 contenant la racine *et cætera*.

1. Écrire la fonction `create_perfect_tree` qui :
   - en entrée prend un `int h` ;
   - en sortie renvoie une valeur de type `list`, avec les numéros des nœuds aux indices correspondants.

   Par exemple `create_perfect_tree(3)` renverra `[None, 1, 2, 3, 4, 5, 6, 7]`

2. Créer les fonctions `left` et `right` qui
   - en entrée prennent un `int i` ;
   - renvoient l'indice du fils gauche (respectivement du fils droit) de `i`.

   À ce niveau on ne se soucie pas des `IndexError`.

3. Implémenter la fonction `prefix` qui devra réaliser l'affichage des nœuds selon un parcours préfixe. (Conseil, passer la liste et l'indice en paramètres et procéder récursivement, les cas d'arrêt se produisant si on voit qu'on va obtenir une `IndexError`).

4. Faire de même avec les fonctions `infix` et `postfix`.