RÉCURRENCE, ARBRES D'INFORMATICIEN

INF303 *TD2*

Récurrence

Exercice 1.

Montrer par récurrence les résultats suivants :

Question 1. La somme des entiers de 1 à n vaut $\frac{n(n+1)}{2}$.

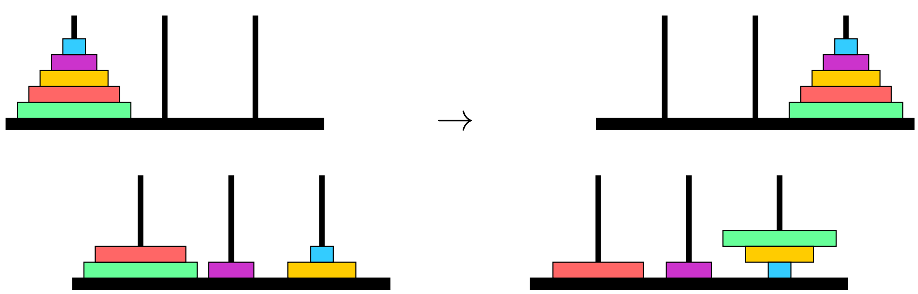
Question 2. $1 + 3 + 5 + \cdots + (2n-1) = n^2$.

Question 3. La somme des n premiers cubes est $\left(\frac{n(n+1)}{2}\right)^2$.

Question 4. $(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$.

Exercice 2. Montrer qu'avec des timbres de 3 et 5 euros, on peut faire des affranchissements pour toutes les sommes entières de plus de 8 euros. (Indice : il faut faire une récurrence à 3 crans).

Exercice 3. Dans le jeu des tours de Hanoi, il y a 3 pics et n disques. Les disques ont toutes des tailles différentes et on supposera que les disques sont dans l'ordre des tailles (le disque 1 est le plus petit, le disque n est le plus grand). Au départ tous les disques sont disposés du plus grand au plus petit sur un des pics. Il faut déplacer tous les disques pour les mettre sur un autre pic. On ne peut deplacer les disques que un par un, on ne peut prendre qu'un disque qui est au dessus, et on ne peut pas mettre un disque sur un disque plus petit.



Autorisée Interdite

Montrer par récurrence qu'il faut au moins $2^n - 1$ coups pour pouvoir déplacer toute la pile.

Arbres d'informaticien : parcours et backtracking

Rappel de vocabulaire :

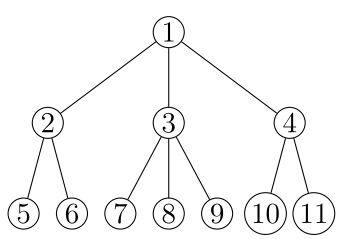
— *Enfants, fils* d'un noeud : noeuds directement reliés en dessous du noeud.
— *Descendants* d'un noeud : tous les noeuds reliés en dessous par un chemin (les enfants, les enfants des enfants,...)
— *Parent, père* d'un noeud : noeud directement relié au dessus (il est unique !)
— *Ancêtres* d'un noeud : tous les noeuds reliés au dessus (le parent, le parent du parent,...)
— *Feuille* : noeud sans enfant
— *Racine* : noeud sans parent
— *Noeud interne* : noeud avec au moins un enfant
— *Profondeur d'un noeud* : distance avec la racine (nombre de traits) (la profondeur de la racine est 0 !)
— *Profondeur (hauteur) de l'arbre* : maximum des profondeurs sur tous les noeuds
— *Largeur* : nombre maximum de noeud sur une même profondeur.
— *Arbre trivial* : ne contient qu'un noeud
— *Arbre k-aire* : tous les noeuds ont au plus k enfants. Attention, l'emplacement des enfants a une importance ! Pour $k = 2$, on parle d'arbre *binaire.*
— *Arbre k-aire complet* : tous les niveaux sont remplis (sauf peut-être le dernier).

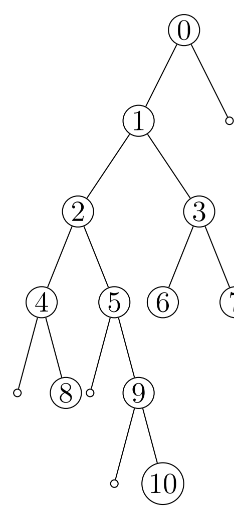
Exercice 4. Dans le cours, on a vu qu'un arbre peut toujours être encodé par un arbre binaire, en suivant la transformation suivante : chaque noeud de l'arbre de départ à comme fils gauche son premier fils, et comme fils droit son premier frère.

Question 1. Transformer l'arbre de gauche en arbre binaire T_B.

Question 2. Comment retrouver la profondeur d'un nœud dans l'arbre gauche en utilisant T_B ?

Question 3. Retrouver l'arbre correspondant à l'arbre binaire de droite.





Exercice 5. Pour l'arbre de la figure 1, donnez l'ordre des sommets pour le parcours préfixe, suffixe et infixe.

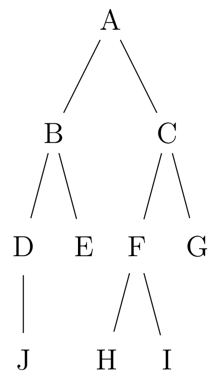
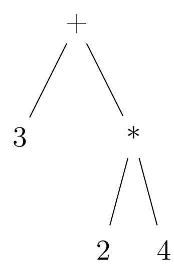


Figure 1 – Parcours d'un arbre

Exercice 6. Donnez les expressions infixe, préfixe et postfixe de l'expression correspondant à l'arbre suivant.



Exercice 7. On rappelle l'algorithme du parcours en largeur. Il utilise deux *couches*, la couche *courante* et la couche des *enfants*.

— au départ, on met la racine dans la couche courante, on prend une liste vide pour la couche des enfants

— ensuite, on parcours les nœuds de la couche courante, en ajoutant leurs enfants dans la couche des enfants

— quand on a terminé le parcours, on change les couches : on prend la couche des enfants comme nouvelle couche courante, et on recommence le parcours.

Quand, à la fin du parcours de la couche courante, on obtient une couche des enfants vide, l'algorithme s'arrête.

Question 1. Appliquer l'algorithme du parcours en largeur sur l'arbre de la figure 1.

Question 2. Utilise-t-on une pile ou file dans cet algorithme ? Que se-passe t-il si on inverse ? Aurait-on eu le même résultat ?

Question 3. Peut-on n'utiliser qu'une seule liste ? Avec quelle structure ?

Exercice 8.

Question 1. Que se passe-t-il si l'on utilise une pile et non pas une file dans l'algorithme du parcours en largeur avec une seule liste ?

Question 2. Comment faire pour calculer en même temps la profondeur des noeuds ?

Exercice 9. Pour un arbre à n nœuds, déterminer (en termes de n) :

1. la plus grande profondeur possible,
2. la plus grande largeur possible,
3. la plus petite profondeur possible,
4. la plus petite largeur possible.

Exercice 10. Pour un arbre binaire à $n = 2^k - 1$ nœuds, déterminer (en termes de n) :

1. la plus grande profondeur possible,
2. la plus grande largeur possible,
3. la plus petite profondeur possible,
4. la plus petite largeur possible.

Exercice 11. Quelle est la largeur d'un arbre k-aire complet en fonction de la profondeur de l'arbre ?

Exercice 12. Reines sur un échiquier

On veut savoir s'il est possible de placer 8 reines sur un echiquier 8×8, de manière à ce qu'aucune paire de reines ne puissent s'attaquer.

Question 1. Donnez une condition nécessaire et suffisante pour que deux reines sur des cases (i, j) et (k, l) ne se mangent pas.

Question 2. Comment décrire un positionnement de 8 reines sur un plateau ? Combien y a-t-il de possibilités ? Peut-on facilement éliminer certains cas ? Combien en reste-t-il ?

Question 3. Décrire l'arbre de backtracking : sa racine, ses noeuds, la relation entre un noeud et ses fils, la condition de blocage.

Question 4. Faire l'arbre pour le cas de 4 reines sur un plateau 4 par 4.

Exercice 13. Les cruches d'eau

Soient deux cruches A et B avec des capacités respectives de 4 et 3 litres. A l'état initial les deux cruches sont vides, et par une série de manipulations on veut obtenir 2 litres d'eau dans la cruche A. Trouvez une solution en utilisant du backtracking.