

ARBRES BINAIRES

Exercice 0 Un tournoi de football est organisé entre 8 équipes. Le tournoi démarre au stade des quarts de finale. L'organisateur affiche les informations suivantes le jour du tournoi :

- En quarts de finale :

quart de finale 1 : Équipe 1 contre Équipe 5
 quart de finale 2 : Équipe 2 contre Équipe 6
 quart de finale 3 : Équipe 3 contre Équipe 7
 quart de finale 4 : Équipe 4 contre Équipe 8

- En demi-finales :

demi-finale 1 : vainqueur quart de finale 1 contre vainqueur quart de finale 3
 demi-finale 2 : vainqueur quart de finale 2 contre vainqueur quart de finale 4

- En finale :

finale : vainqueur demi-finale 1 contre vainqueur demi-finale 2

Propose une représentation arborescente permettant de mieux visualiser ces informations.

Exercice 1 On s'intéresse au jeu suivant (famille des jeux de Nim) :

Règles

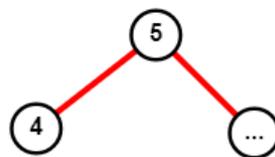
On dispose de 5 allumettes. Chacun son tour, chaque joueur peut en prendre une ou deux. Celui qui prend la dernière gagne.

L'état du jeu est complètement déterminé par le nombre d'allumettes restantes et le nom du prochain joueur à jouer.

On peut utiliser un arbre pour représenter toutes les parties possibles en considérant que :

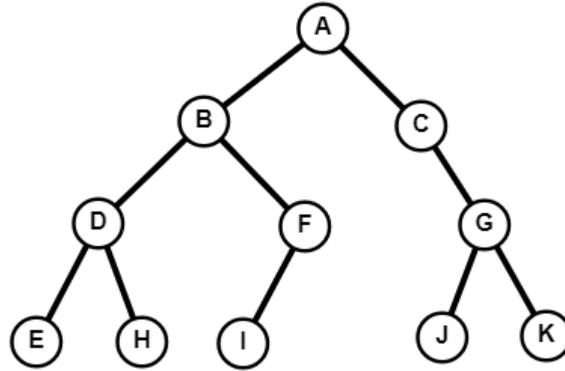
- la valeur d'un noeud est le nombre d'allumettes restantes
- la couleur de chaque arête correspond au joueur qui doit jouer (rouge ou bleu, c'est le rouge qui commence)

Voici le début de l'arbre :



1. Recopie et complète cet arbre.
2. Pourquoi est-ce un arbre binaire ?
3. Combien de possibilités de gagner ont chacun des deux joueurs ? Expliquez.
4. Trouve une stratégie assurant la victoire au premier joueur (rouge) dans tous les cas ? Explique.

Exercice 2 On considère l'arbre binaire suivant :

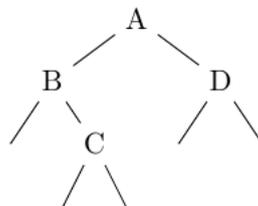


1. Quelle est la taille de cet arbre ?
2. Liste les noeuds internes puis les feuilles.
3. Quelle est la hauteur de cet arbre ?
4. Quels sont les fils du noeud B ?
5. Quel est le sous-arbre gauche du noeud B ?
6. Quel est le sous-arbre gauche du noeud C ?

Exercice 3 Écris une fonction `affiche(a)` qui imprime un arbre sous la forme suivante :

- pour un arbre vide, on n'imprime rien ;
- pour un noeud, on imprime une parenthèse ouvrante, son sous-arbre gauche (récursivement), sa valeur, son sous-arbre droit (récursivement), puis enfin une parenthèse fermante.

Par exemple l'arbre



sera affiché sous la forme `((B(C))A(D))`

Exercice 4 Dessine l'arbre binaire sur lequel la fonction `affiche` de l'exercice précédent produit la sortie `(1((2)3))`.

Exercice 5 Ajoute à la classe `Noeud` une méthode `__eq__` permettant de tester l'égalité entre deux arbres binaires à l'aide de l'opération `==`. Attention à bien prendre en compte le cas de l'arbre vide.

Exercice 6 Écrire une fonction `parfait(h)` qui reçoit en argument un entier `h` supérieur ou égal à zéro et renvoie un arbre binaire parfait de hauteur `h`.

Précision : Un arbre binaire parfait est un arbre binaire où toutes les feuilles sont exactement à la même profondeur.