ARBRES BINAIRES

0. Introduction

Exercice 1 : Tournoi à élimination directe.

Dans ce type de compétition, un joueur est éliminé au premier match qu'il perd. Le vainqueur est celui qui gagne tous ses matches.

PARTIE A : On suppose qu'il y a 16 joueurs

1. Représenter cette situation à l'aide d'un arbre.

1. En combien de tours se déroule ce tournoi ?

1. Combien de matches au total seront joués pour désigner un vainqueur ?

PARTIE B : On suppose qu'il y a N joueurs.

1. Quelle condition peut-on envisager sur N pour que le tournoi soit simple à organiser ?

1. Sous cette condition :
    - En combien de tours se déroule ce tournoi ?

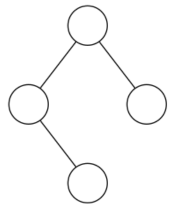
- Combien de matches au total seront joués pour désigner un vainqueur ?

1. Définition

La situation précédente se décrit à l'aide d'un arbre binaire. Un arbre binaire est un cas particulier de structure arborescente ou chaque noeud à au maximum deux enfants. Plus précisément, un arbre binaire est un ensemble fini de noeuds correspondant à l'un des cas suivants:

- Soit l'arbre est vide (il ne contient aucun noeud)
- Soit l'arbre n'est pas vide, et ses noeuds sont structurés de la façon suivante :
  - L'un des noeuds est la racine de l'arbre.
  - Les noeuds restants sont séparés en deux sous ensembles, qui forment récursivement deux sous-arbres appelés respectivement sous-arbre gauche et sous-arbre droit.
  - La racine est reliée à ses deux sous-arbres gauche et droit, plus précisément à la racine de chacun des sous-arbres (s'ils ne sont pas vides)

Exemple :



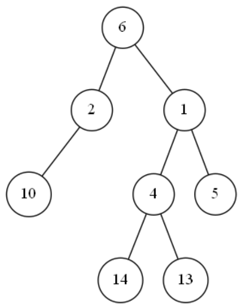
- Dans l'exemple ci-contre, l'arbre binaire comporte 4 noeuds.
- La racine est en haut.
- Ici le sous-arbre gauche contient 2 noeuds et le sous-arbre droit contient 1 noeud.

Remarques :

- Bien que l'on ne représente pas de noeud lorsque l'arbre est vide, il est important de garder en mémoire que chaque noeud possède exactement 2 sous-arbres (qui peuvent être vides) qu'il faut distinguer (même s'il n'y en a qu'un) :
  - Un sous-arbre gauche
  - Un sous-arbre droite
- On utilise le même vocabulaire que pour les structures arborescentes : racine, enfant, feuille, taille, hauteur.
- L'intérêt des arbres est d'y stocker de l'information (valeurs alphanumériques, opérateurs, balises html, etc...).

Exercice 2:

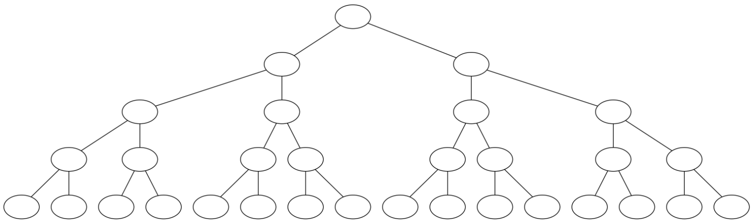
Compléter les phrases ci-dessous :



- Dans l'exemple ci-contre, la taille de l'arbre binaire est __.
- Sa hauteur est __.
- La racine est __.
- ___ est une feuille (il possède deux sous-arbres vides)
- ___ est le sous-arbre gauche de l'arbre 2
- L'arbre 1 possède deux enfants qui sont les sous-arbres _ .
- ___ est le sous arbre gauche de l'arbre 1
- La racine du sous-arbre droit de ___ est 1

Exercice 3 :

Reprenons l'arbre correspondant à un tournoi à élimination directe avec 16 joueurs. Dans cet arbre, chaque noeud possède exactement 2 enfants, on parle d'arbre binaire parfait



PARTIE A :

1. Quelle est la hauteur de l'arbre ?

1. Quelle est la taille de l'arbre ?

1. Combien de feuilles possède-t-il ?

PARTIE B : On suppose désormais qu'il y a N joueurs et que N est une puissance de 2, c'est à dire que $N = 2^n$ avec n entier naturel.

1. Quelle est la hauteur de l'arbre ?

1. Quelle est la taille de l'arbre ?

1. Combien de feuilles possède-t-il ?

Remarques :

- Si h est la hauteur d'un arbre binaire, alors :
  - La taille maximale de l'arbre est $2^h - 1$ (chaque noeud possède exactement 2 enfant)
  - La taille minimale de l'arbre est h (chaque noeud possède exactement 1 enfant).

- Autrement dit, la taille N d'un arbre binaire de hauteur h vérifie $h \leq N \leq 2^h - 1$

Exercice 4 :

1. Quelle est la taille maximale d'un arbre binaire de hauteur 5 ?

1. Donner un encadrement de la hauteur d'un arbre binaire dont la taille est 8.Pour chaque valeur, dessiner un arbre binaire correspondant

Exercice 5 :

En n'oubliant pas qu'il y a des sous-arbres gauches et sous-arbres droits, dessiner tous les arbres binaires ayant :

1. 3 noeuds

1. 4 noeuds

Exercice 6:

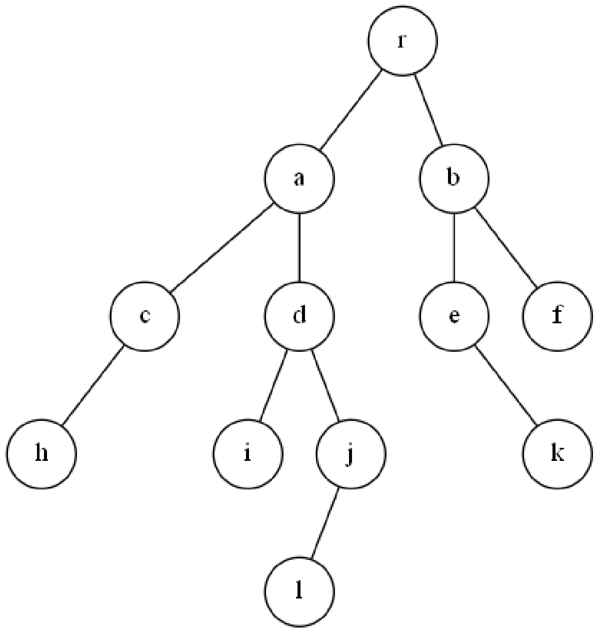
En complément de l'exercice précédent, sachant qu'il y a 1 arbre binaire vide, 1 arbre binaire contenant 1 noeud , 2 arbres contenant binaires 2 noeuds, 5 arbres binaires contenant 3 noeuds, 14 arbres binaires contenant 4 noeuds, dénombrer le nombre d'arbres binaires à 5 noeuds (on ne cherchera pas à les représenter tous...).

2. Parcours

- Parcourir un arbre c'est partir d'un noeud et visiter tous les noeuds de l'arbre une seule fois.
- Ce concept de parcours est très important en algorithmique.
- Les parcours permettent notamment de générer une autre représentation de l'arbre(par exemple une liste) ou d'effectuer une recherche dans une structure arborescente.
- Revoyons les différents parcours d'arbres, et définissons un nouveau parcours pour les arbres binaires.

Exercice 7 :

Considérons l'arbre ci-contre. Pour chaque définiton de parcours, indiquer l'ordre dans lequel les noeuds sont visités.



1. Parcours en largeur : les noeuds sont parcourus par profondeur croissante (disons de gauche à droite).

Réponse :

1. Parcours en profondeur préfixe : On effectue le traitement de chaque noeud avant d'explorer le sous-arbre gauche puis le sous-arbre droit.

$parcours\_profondeur\_prefixe(noeud)$:
$\quad traitement(etiquette)$
$\quad pour\ chaque\ enfant\ de\ noeud$:
$\qquad parcours\_profondeur\_prefixe(enfant)$

Réponse :

1. Parcours en profondeur suffixe(ou postfixe) : On effectue le traitement de chaque noeud après avoir exploré le sous-arbre gauche puis le sous-arbte droit.

$parcours\_profondeur\_postfixe(noeud)$:
$\quad pour\ chaque\ enfant\ de\ noeud$:
$\qquad parcours\_profondeur\_postfixe(enfant)$
$\quad traitement(etiquette)$

Réponse :

1. parcours en profondeur infixe (propres aux arbres binaires) : On liste chaque noeud ayant un fils gauche la seconde fois qu'on le voit et chaque noeud sans fils gauche la première fois qu'on le voit. En d'autres termes on traite, le sous arbre gauche, puis la racine, puis le sous-arbre droit.

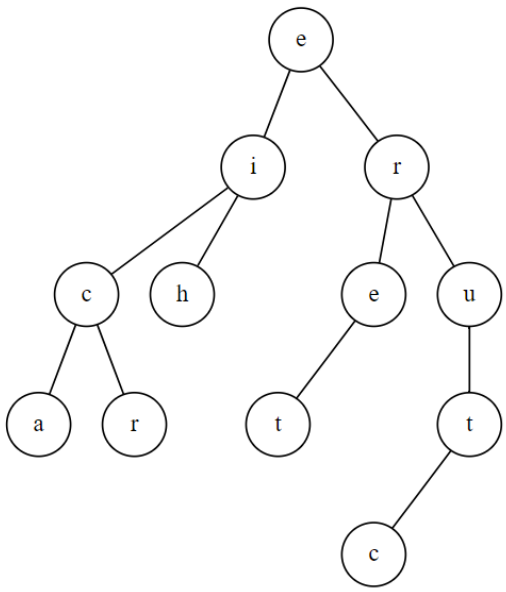
Réponse :

Remarques :

- Quelques applications des arbres binaires :
  - Les arbres binaires fournissent un modèle de représentation utilisé efficacement lors de recherches d'information.
  - Ils sont aussi utilisés lors de la transformation d'un programme écrit en langage évolué(comme python) en une suite d'instructions élémentaires exécutables par une machine.
  - Le code Morse(composé de `.` et de `-`) peut être représenté par un arbre binaire.
  - Le codage de Huffman qui permet de compresser des données sans perte utilise un arbre binaire.
  - L'algorithme dit de "tri rapide" (ou tri du bijoutier) repose sur un arbre binaire.

Exercice 8 :

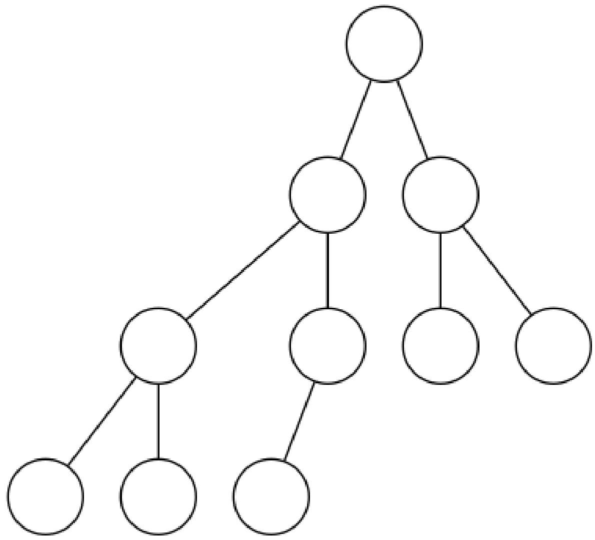
Trouver le mot caché dans l'arbre binaire ci-contre à l'aide d'un parcours. De quel parcours s'agit-il ?



Exercice 9 :

Voici le message obtenu après un parcours infixe de l'arbre ci-contre : `gouvernail` .

Retrouver la lettre contenue dans chaque noeud.



In [ ]: