SDD_Arbres

I/ Exemples d'arbres

Exemple : organisation d'un tournoi de rugby

Un organisateur de tournoi de rugby recherche la meilleure solution pour afficher les potentiels quarts de final, demi-finales et finale :

Au départ nous avons 4 poules de 4 équipes. Les 4 équipes d'une poule s'affrontent dans un mini championnat (3 matchs par équipe). À l'issue de cette phase de poule, les 2 premières équipes de chaque poule sont qualifiées pour les quarts de finale.

Dans ce qui suit, on désigne les 2 qualifiés par poule par :

- Poule 1 => 1er Eq1 ; 2e Eq8
- Poule 2 => 1er Eq2 ; 2e Eq7
- Poule 3 => 1er Eq3 ; 2e Eq6
- Poule 4 => 1er Eq4 ; 2e Eq5
-

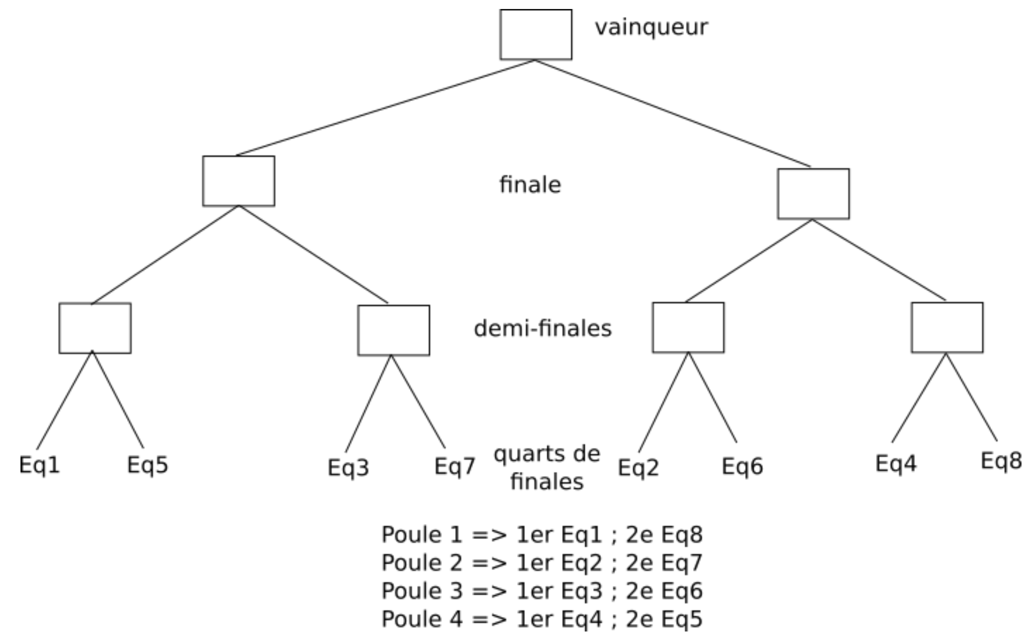
En quart de final on va avoir :

- quart de finale 1 => Eq1 contre Eq5
- quart de finale 2 => Eq2 contre Eq6
- quart de finale 3 => Eq3 contre Eq7
- quart de finale 4 => Eq4 contre Eq8

Pour les demi-finales on aura :

- 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

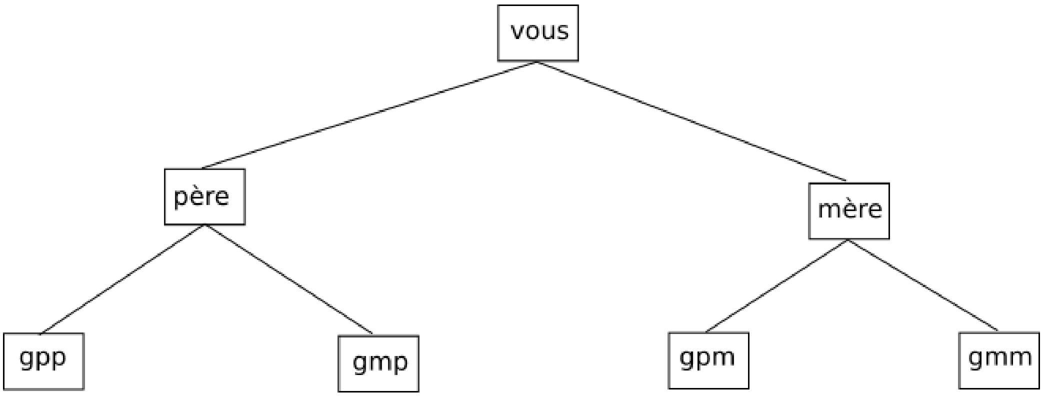
L'organisateur du tournoi affiche les informations ci-dessus le jour du tournoi. Malheureusement, la plupart des spectateurs se perdent quand ils cherchent à déterminer les potentielles demi-finales (et ne parlons pas de la finale !)
Pourtant, un simple graphique aurait grandement simplifié les choses :



Les spectateurs peuvent alors recopier sur un bout de papier ce schéma et ensuite se livrer au jeu des pronostics. Nous avons ci-dessous ce que l'on appelle une structure en arbre.

Exemple : un arbre généalogique

On peut aussi retrouver cette même structure dans un arbre généalogique :



Ces deux exemples peuvent être représentés sous forme **d'arbres**. On pourrait citer comme autres exemples l'organisation de fichiers dans un ordinateur.

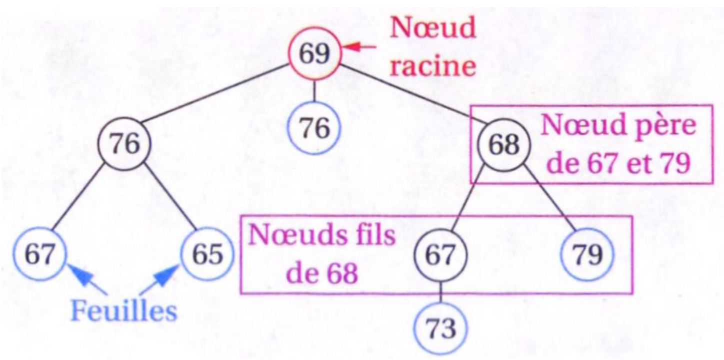
I/ Notion d'arbres

1/ Définition d'un arbre enraciné

Un arbre enraciné (ou arborescence) est constitué de nœuds organisés de manière hiérarchique : c'est un **graphe non orienté (*), connexe(*), sans cycle(*)** dans lequel on a choisi un **nœud de départ**, la **racine**.

Chaque **nœud** est en général étiqueté par une information et a **un seul nœud père** (sauf la racine bien sûr).

Un exemple d'arbre



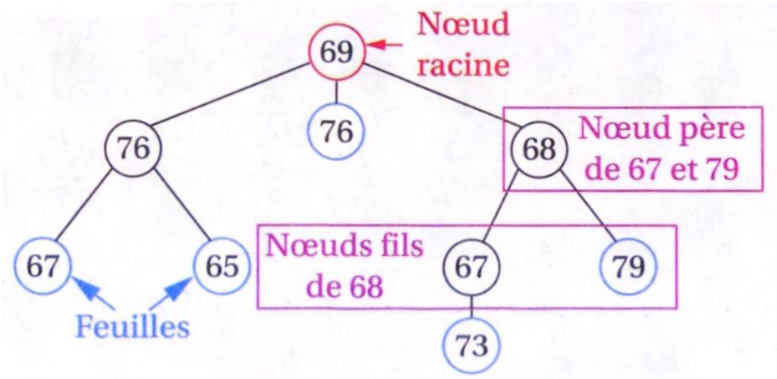
Graphe non orienté : on peut aller d'un nœud à un autre dans les deux sens.

Graphe connexe : chaque nœud est relié à au moins un autre nœud.

Graphe sans cycle : il est impossible de revenir à un nœud par un chemin comportant au moins 3 nœuds sans repasser plusieurs fois par un des nœuds traversés. Cela est typique des graphes hiérarchiques.
Il faudrait sur l'exemple un lien entre 65 et 67 pour avoir un cycle partant de 69 : 69 ->76->65->67->68->69.

2/ Caractéristiques d'un arbre

Exemple d'arbre



A savoir : Les définitions suivantes sont à connaître !

- **Taille d'un arbre : nombre de nœuds d'un arbre** (9 dans l'exemple précédent)**.**
- **Feuille : nœud n'ayant pas de fils. On les appelle aussi nœud externe, les autres étant des nœuds internes.**
- **Arité : nombre de fils d'un nœud. Deux nœuds ayant le même père sont dits nœuds frères** (l'arité de 69 est de deux et 76 / 68 sont des nœuds frères).
- **Profondeur d'un nœud : longueur du chemin le plus court vers la racine** (la profondeur de 67 est de 2 par exemple). **Celle de la racine est donc de zéro (on rencontre parfois une profondeur de 1 dans la littérature).**
- **Hauteur d'un arbre : profondeur la plus élevée possible** (ici c'est 73 qui a la plus grande profondeur soit 3).

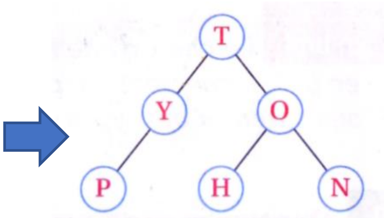
III/ Arbres binaires

1/ Définitions

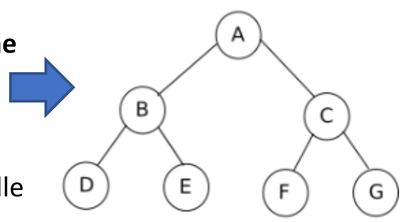
A savoir : Les définitions suivantes sont à connaître !

Arbre binaire : arbre dont chaque nœud a au plus deux fils. On peut parler de sous-arbre à droite et à gauche.

Arbre équilibré : arbre pour lequel chaque nœud interne, les sous-arbres droit et gauche ont la même hauteur



Arbre binaire complet : arbre pour lequel chaque nœud interne a exactement deux fils.

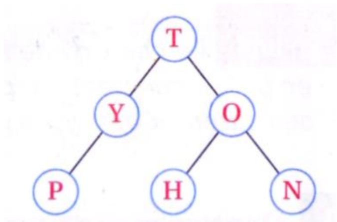


Remarque : les notions d'arbres « équilibré » et « binaire complet » ont des définitions variables, il faut être attentif à celle qui est donnée !

2/ Parcours d'un arbre binaire

A savoir : Les définitions suivantes sont à connaître !

On s'appuiera sur l'exemple à droite pour illustrer les définitions suivantes.



Un **parcours d'un arbre** définit dans quel ordre on parcourt ses nœuds :

Parcours en largeur (à connaître)

Les nœuds sont parcourus « étage par étage » : la racine, puis ses fils, de gauche à droite.
Sur l'exemple, on obtiendrait : T ; Y ; O ; P ; H ;N

A noter : ce type de parcours est relativement rare.

Parcours en longueur (à connaître)

L'un des sous-arbres est complètement exploré avant que le parcours du second ne commence. On distingue trois types de parcours :

- **Le parcours préfixe (ou préordre) : chaque nœud est visité avant que ses fils le soient. On part de la racine puis on visite le fils à gauche jusqu'à ses feuilles puis le fils droit.**
  Sur l'exemple, on aurait : T ; Y ; P ; O ; H ; N
- **Le parcours infixe (ou ordre) : chaque nœud est visité après son fils gauche mais avant son fils droit (fils gauche, nœud père (racine éventuellement), fils droit).**
  Sur l'exemple, on aurait : P ; Y ; T ; H ; O ; N
- **Le parcours postfixe (ou postordre) : chaque nœud est visité après que ses fils le soient (fils gauche, fils droit, racine).**
  Sur l'exemple, on aurait : P ; Y ; H ; N ; O ; T

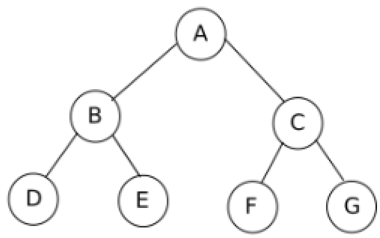
A noter : le parcours « préfixe » est peu utilisé du fait du parcours des arbres par récursivité 😊.

3/ Relation entre hauteur d'un arbre binaire et nombre de nœuds

On considère un arbre binaire complet ayant *n* nœuds.

A savoir : l'arbre binaire complet a une hauteur de la partie entière de $\log_2(n)$.

Dans l'exemple, il y a 7 nœuds et $\log_2(7)$ vaut environ 2,8 soit 2 pour la partie entière.



Remarque : on peut ramener cette relation à celle liant « nombre de bits nécessaire » et « représentation des entiers naturels ».

- 1 bit pour représenter 0 et 1 (ici les nœuds « B » et « C »).
- 2 bits pour représenter 0, 1, 2 et 3 (ici les nœuds « D », « E », « F » et « G ».

A noter : la pire configuration serait représentée par un arbre sous forme « filiforme » ou « dégénéré ». On aurait une hauteur de valeur $n - 1$.

Sur l'exemple précédent, l'arbre a une taille de quatre et une hauteur de trois.