Algorithmique

TD 3 : Parcourir & s'Habiller

Exercice 1 (Représentations de Graphes) Donnez trois représentation (informatiques) d'un graphe, ainsi que les algorithmes permettant de passer d'une représentation à une autre.

Exercice 2 (Parcours Classiques d'Arbres) Il existe trois parcours classiques sur les arbres (binaires) :

Parcours Pré-Ordre

```
Si l'Arbre est Non-Vide
    Examen de la Racine
    Pré-Ordre (Fils Gauche)
    Pré-Ordre (Fils Droit)
```

Parcours Post-Ordre

```
Si l'Arbre est Non-Vide
    Post-Ordre (Fils Gauche)
    Post-Ordre (Fils Droit)
    Examen de la Racine
```

Parcours En-Ordre

```
Si l'Arbre est Non-Vide
    En-Ordre (Fils Gauche)
    Examen de la Racine
    En-Ordre (Fils Droit)
```

Que donnent ces trois parcours sur l'arbre de la figure 1 & sur un arbre de recherche ?

Exercice 3 (Parcours de Graphes) On a souvent besoin de *parcourir* un graphe, c'est-à-dire, en partant d'un sommet, & en passant chaque fois d'un sommet à un de ses voisins, d'"explorer/marquer" tous les sommets du graphe. Il existe pour cela deux stratégies classiques :

En Largeur d'Abord (BFS) En partant d'un sommet x, on marque tous ses voisins (les sommets à distance 1), puis tous les voisins non marqués des voisins de x (les sommets à distance 2), ...

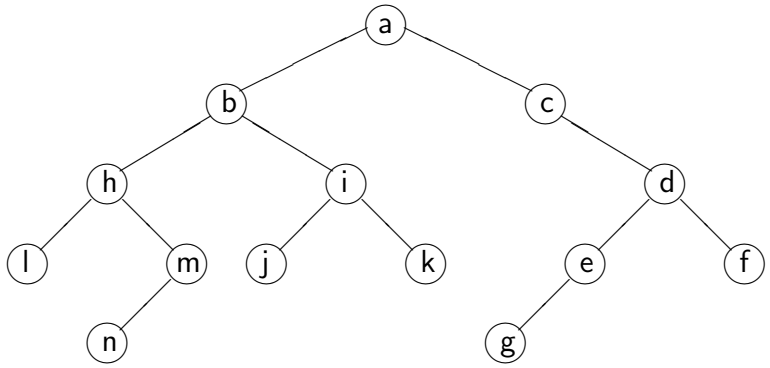

Figure 1: Un arbre

En Profondeur d'Abord (DFS) En partant d'un sommet x_0, on va sur un sommet x_1 (voisin de x_0), puis sur un sommet non marqué x_2 voisin de x_1, ... Si on est coincé (i.e. on est en un sommet x_k dont tous les voisins sont marqués, & tous les sommets n'ont pas été marqués), on revient en arrière jusqu'à ce qu'il soit possible de recommencer.

Donnez les algorithmes correspondant à ces parcours.

Exercice 4 (Habillage) On a souvent besoin aussi de s'habiller. Représentez les contraintes de l'habillage (i.e. *il faut mettre les chaussettes avant les chaussures*) par un graphe orienté. Montrez qu'un parcours en profondeur d'abord sur ce graphe apporte une aide à l'habillage.

Exercice 5 (Autres Parcours d'Arbres) Donnez un algorithme qui parcourt les sommets d'un arbre "étage par étage" & un algorithme qui parcourt les sommets du graphe de la figure 1 par ordre alphabétique.