

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 9

23 Novembre 2007

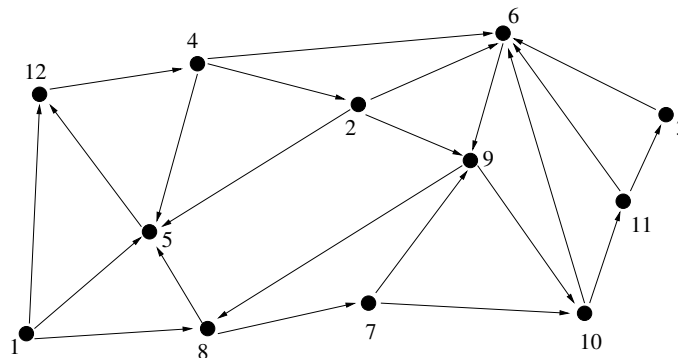
1. Recherche dans un labyrinthe

On aimerait utiliser l'algorithme DFS du cours pour trouver la sortie d'un labyrinthe. Notre modèle de labyrinthe est un graphe dont les sommets sont des chambres et les arêtes sont des portes entre les chambres. L'un des nœuds est le départ, et un autre nœud est la sortie.

- Donner une variante de l'algorithme DFS qui donne explicitement le chemin des nœuds visités. On précise que simplement voir si un nœud est marqué n'est pas considéré une visite de ce nœud. *Indice* : En descendant, il faut mémoriser le retour à prendre afin d'obtenir une suite de sommets adjacents.
- Montrer qu'une arête quelconque apparaît au plus deux fois sur le chemin.
- Montrer que le chemin pour trouver la sortie est de longueur au plus $2n - 3$.
- Donner une famille de graphes (pour des n arbitrairement grands) avec un sommet de départ tel que tout chemin de longueur $2n - 4$ omet au moins un sommet.
- Montrer que dans un graphe quelconque, tout chemin de longueur $< n - 1$ omet au moins un sommet.

2. Traverser des graphes

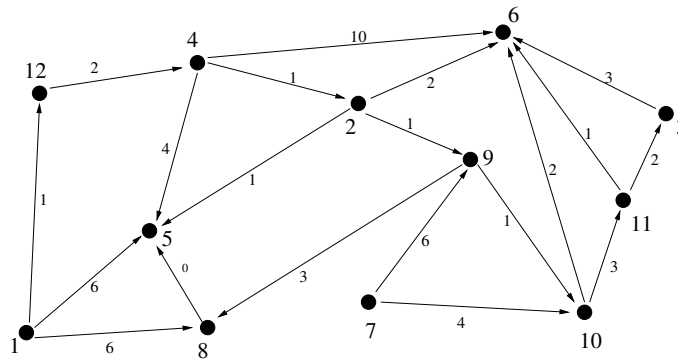
Voici un graphe orienté :



- Parcourir ce graphe avec la méthode DEPTHFIRSTSEARCH, en commençant au sommet 1.
- Parcourir ce graphe avec la méthode BREADTHFIRSTSEARCH, en commençant au sommet 1.
- Modifier l'algorithme BFS pour qu'il retourne pour chaque sommet sa distance par rapport à 1.
- Modifiez l'algorithme ABSTRACTTRAVERSAL pour qu'il compte aussi le nombre de sommets qui sont atteignables depuis 1.

3. L'algorithme de Dijkstra

Considérer le graphe suivant :



- a) Appliquer l'algorithme de Dijkstra sur ce graphe. Commencer au sommet 1.
- b) Modifier l'algorithme de Dijkstra pour qu'il fournisse pour chaque sommet v le plus court chemin du sommet de départ s à v et l'ensemble des sommets qui font partie de ce chemin.

4. Dijkstra et Programmation Dynamique

- a) Comment peut-on se servir de l'algorithme de Dijkstra pour résoudre le problème du Knapsack 0/1 ?
- b) Utiliser l'algorithme de Dijkstra pour résoudre le problème LCS. (Donner la construction du graphe.)
- c) Comparer dans chacun des cas ci-dessus le nombre d'arêtes du graphe obtenu avec le temps de parcours de l'algorithme classique DP pour résoudre le problème.
- d) Peut-on utiliser l'algorithme de Dijkstra pour résoudre le problème de disposition optimale de parenthèses ?

5. Dijkstra et Moore-Bellman-Ford

On considère un graphe $G = (V, E)$ avec des poids p_i sur les arêtes ($i \in E$), et deux sommets s et t .

- a) Si l'on multiplie les poids par une constante, le plus court chemin de s à t reste-t-il le même ?
- b) Si l'on ajoute aux poids une constante c , le plus court chemin de s à t , reste-t-il le même ?
- c) Si les poids des arêtes sont non-négatifs, on peut appliquer ou bien Dijkstra ou bien Moore-Bellman-Ford. Lequel des algorithmes est alors préférable ? Donner une famille de graphes tel que si le nombre de sommets n croît, l'un des algorithmes est $O(1)$ tandis que l'autre est $O(n^3)$.