

TP Graphes / parcours et labyrinthe

1 Parcours en profondeur

Q 1 . Ecrivez une procédure `ParcoursEnProfondeur` qui prend en entrée un graphe non orienté et un sommet de départ, et qui affiche la liste des sommets parcourus en profondeur d'abord. Vous ferez un parcours en vous servant des voisins, ce qui implique que votre procédure fonctionnera aussi bien pour des graphes orientés que non orientés. Vous utiliserez la version efficace, c'est-à-dire la version donnée page 71 du poly.

Q 2 . Testez votre procédure sur un exemple simple.

2 Sortie du labyrinthe

Faire un parcours en profondeur est la méthode naturelle pour sortir d'un labyrinthe (plutôt qu'un parcours en largeur). Saurez vous dire pourquoi ? En pratique, comment fait-on un parcours en profondeur d'un labyrinthe ?

Q 3 . Ecrire une fonction (en vous inspirant du parcours en profondeur) qui décide si oui ou non il existe une chaîne entre deux sommets.

Q 4 . Utilisez votre fonction pour savoir de quels labyrinthes (voir les fichiers `labyrinthe*.grp`) on peut sortir. Dans chacun, les entrées et sorties sont les sommets nommés `entree` et `sortie`.

Q 5 . Modifiez votre fonction pour qu'elle calcule en même temps un chemin menant de l'entrée à la sortie.

Q 6 . Trouvez un labyrinthe sur Internet, et codez le en graphe pour trouver sa (ou une de ses) solution(s).