

1. Principaux objectifs

- Savoir définir et reconnaître différents types de labyrinthe :
 - parfait
 - non parfait
- Savoir faire le lien labyrinthe \leftrightarrow graphe.
C'est un arbre dans le cas d'un labyrinthe parfait.
- Savoir mettre en œuvre un algorithme de recherche de la sortie dans un labyrinthe parfait.
Lien avec le parcours sur l'arbre.
- Comprendre l'algorithme de Prim de génération de labyrinthe parfait. Proposer une amélioration de l'algorithme pour améliorer la « qualité » du labyrinthe
- L'algorithme de Prim dans un autre contexte \leftrightarrow : recherche d'un arbre couvrant minimal.
Exemple d'un réseau électrique.

2. Modalités

a) Séance 1

- Des exemples de labyrinthes (parfaits ou non),
- Un exemple de génération : aléatoire \diamond
- Graphes correspondants,
- Parcours ?

b) Séance 2

Calcul de qualité d'un labyrinthe parfait.

- nombre de murs, ★
- Plus longue ligne droite, ★
- Nombre de virages... ★
- Nombre de cul-de-sac. ★

c) Séance 3

Suite qualité. Algorithme de Prim

d) Séance 4

Algorithme de Prim amélioré

e) Séance 5

Application à la recherche d'un arbre couvrant minimal (ex d'un réseau électrique)

3. Resources

culturelle <http://fr.wikipedia.org/wiki/Labyrinthe>

informatique

http://fr.wikipedia.org/wiki/Mod%C3%A9lisation_math%C3%A9matique_d%27un_labyrinthe

Tangente n° 57

Tangente Hors Série n° 12