

Projet Labyrinthe



INTRODUCTION

Le but de ce projet est d'étudier les *labyrinthes* en tant qu'objets mathématiques et de chercher leurs différentes propriétés.

Mots clés : combinatoire, énumération, algorithmes.

Le principe du projet est d'étudier une question mathématique complexe de façon autonome et originale en utilisant l'outil informatique. **L'ensemble du projet est à faire par groupe de 3 étudiants.** Le projet est divisé en 2 sections : théorique et pratique. La première partie est **obligatoire**, la seconde partie correspond à des **pistes de travail**.

Rendu du projet :

- Un **rapport de mi projet** répondant aux questions de la Section 1.
- Un **exposé** présentant votre travail et vos résultats sur la Section 2 en fin de projet.

Le Rapport. .

Date de rendu : 29 mars à minuit.

Que faut-il faire ? Répondre aux questions de la Section 1.

Quel format ? Format PDF obligatoire. Si vous souhaitez ajouter des dessins, vous pouvez les scanner et les ajouter à votre document.

Nom du fichier : labyrinthe_nom1_nom2_nom3.pdf

Qui réalise le rapport ? Un rapport par groupe de 3 étudiants.

A qui le rendre ? Envoyez par mail à viviane.pons@u-psud.fr, objet du mail "Rapport Projet labyrinthe" et noms des étudiants dans le corps du mail.

L'exposé. .

Quand ? Les exposés auront lieu en fin de semestre, la date sera précisée ultérieurement.

Que doit-on présenter ? Vous devez présenter le travail effectué sur la Section 2 du projet, en particulier, les résultats que vous avez obtenus, les algorithmes que vous avez utilisés, les images ou vidéos produites, etc.

En combien de temps ? Vous aurez 10 minutes de présentation, puis 5 minutes de questions.

Sur quel support ? Vous aurez un vidéo projecteur et un ordinateur à disposition (ou le votre si vous le souhaitez). Vous pourrez donc présenter votre exposé sous forme d'un powerpoint ou pdf. Vous pouvez aussi montrer des images, vidéos, démos de code.

Qui parle ? Tout le monde ! Les 3 étudiants doivent participer.

Doit-on présenter notre code ? Vous pouvez utiliser un notebook SageMathCloud pour présenter des démos de code, cependant, nous ne notons pas la programmation mais bien les résultats obtenus !

Doit-on répondre aux questions ? Les questions de la Section 2 ne sont pas obligatoires, ce sont des pistes de travail, vous pouvez les suivre, ou pas...

1. ÉTUDE THÉORIQUE DU PROBLÈME

1.1. **Définitions.** On considère une grille rectangulaire de taille $n \times m$. Le bord du rectangle est appelé *enceinte*, chaque case est appelée une *cellule*. Des *murs* peuvent être placés sur le côté des cellules. On appelle un ensemble *enceinte+murs* un *pseudo-labyrinthe*, voir les exemples Figure 1.

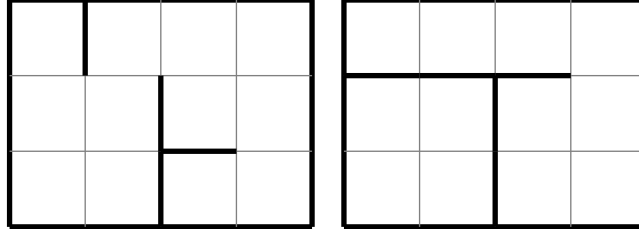


FIGURE 1. Deux exemples de pseudos-labyrinthes de taille 4×3

Pour être un *labyrinthe*, un pseudo-labyrinthe doit vérifier deux conditions :

- (1) l'espace à l'intérieur de l'enceinte est connexe : il existe toujours un chemin entre deux cellules données,
- (2) si l'on rajoute un mur où que ce soit, alors il perd sa connexité.

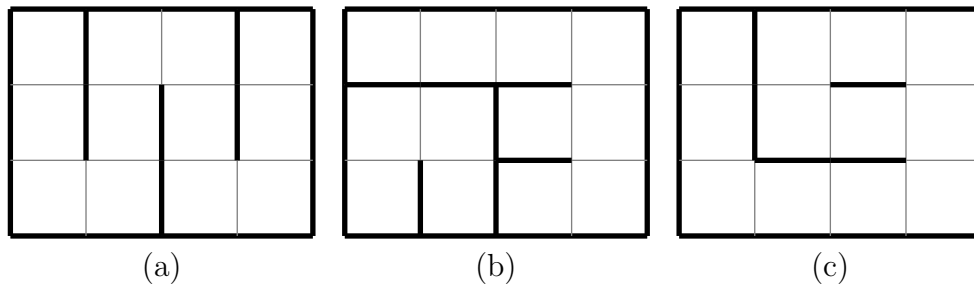


FIGURE 2. Exemple et contre-exemples de labyrinthes

Dans la figure 2, seul (a) est un labyrinthe. En effet, (b) ne satisfait pas la condition (1) et (c) ne vérifie pas la condition (2).

1.2. Questions.

- (1) Dessinez tous les pseudo-labyrinthes de taille 2×2 (vous devez en trouver 16). Combien sont des labyrinthes ?
- (2) Quel est le nombre maximal de murs dans un pseudo-labyrinthe de taille $n \times m$?
- (3) Combien existe-t-il de pseudo-labyrinthes de taille $n \times m$?
- (4) Dessinez tous les labyrinthes de taille 3×2 . Vous devez en trouver 15.
- (5) En comptant les murs de chacun de ces labyrinthes, que remarquez-vous ?
- (6) Soit L un labyrinthe, on suppose que la sortie se trouve dans la cellule en haut à gauche. Prouvez que si je choisis une cellule c quelconque de L , alors il existe un unique chemin sans demi-tours entre c et la sortie (il faut prouver que ce chemin existe et qu'il est unique).
- (7) En utilisant le résultat précédent, il est possible de prouver que pour un labyrinthe de taille n , il existe exactement $n \times m - 1$ bordures entre cellules qui ne sont pas des murs (on ne vous demande pas la preuve). Un pseudo-labyrinthe possédant le "bon" nombre de murs est-il toujours un labyrinthe ? Conjecturez une condition nécessaire et suffisante pour qu'un pseudo-labyrinthe soit un labyrinthe.

2. MODÉLISATION ET EXPLORATION

Le problème mathématique que nous avons présenté ici sous forme de labyrinthes est un véritable problème de recherche qui pose de nombreuses questions. Certaines de ces questions n'ont pas de réponses à ce jour ! Le but de ce projet est maintenant d'explorer ces questions par vous même en utilisant l'outil informatique. Il n'y a pas de solution unique, ni même de direction imposée, nous vous proposons simplement des pistes...

2.1. Modélisation et visualisation. La première étape consiste à représenter les objets "pseudo-labyrinthe" et "labyrinthes" par des objets informatiques.

- (1) Quelle structure de donnée pouvez-vous utiliser pour représenter un pseudo-labyrinthe ?
- (2) Comment pouvez-vous tester si un pseudo-labyrinthe est un labyrinthe ? (Et quelle est la complexité de l'algorithme)
- (3) Implantez une méthode pour "dessiner" un pseudo-labyrinthe représenté par votre structure de donnée (utilisez la méthode *plot*).

Quelques pistes : les tableaux et matrices vous sembleront peut-être les objets les plus évidents, vous pouvez aussi regarder du côté des *graphes*.

2.2. Énumération. Le but de cette section est d'étudier la question du nombre de labyrinthes pour des valeurs de n et m données.

- (1) Proposez et implantez un algorithme pour générer l'ensemble des labyrinthes d'une taille $n \times m$ donnée.
- (2) Vérifiez par des tests sur différentes valeurs les résultats prouvés et conjecturés dans la partie théorique.
- (3) Affichez dans un tableau à double entrée le nombre de labyrinthes de taille $n \times m$ en faisant varier n et m (voyez comment évoluent les nombres et jusqu'où vous pouvez aller). Chaque ligne (ou chaque colonne) du tableau vous donne une liste de nombre que vous pouvez étudier.
- (4) Existe-t-il d'autres moyens de calculer ces nombres que par la génération de tous les labyrinthes ? Intéressez vous aux notions d'arbres couvrants (ou *spanning trees*) et aux algorithmes existants et implantés dans Sage.
- (5) On joue à "quel est le prochain nombre" : comprendre la logique d'une suite de nombres et "deviner" la suite. Pour cela, on peut utiliser le site <http://www.oeis.org> qui référence les suites de nombres connues. Cherchez sur ce site les différentes formules existantes pour calculer les nombre du tableau.
- (6) La première ligne intéressante du tableau est celle pour $n = 2$. Dans ce cas la formule récursive est relativement simple (cherchez la sur <http://www.oeis.org>). Vous pouvez essayer de la "comprendre" : prenez un labyrinthe de taille 2×3 de votre choix et rajouter 2 cellules pour obtenir un pseudo-labyrinthe de taille 2×4 , comment en faire un labyrinthe ? Comment obtenir tous les labyrinthes de taille 2×4 à partir des labyrinthes de taille 2×3 ?

2.3. Génération aléatoire. Réfléchissez à des algorithmes pour générer des pseudo-labyrinthes et labyrinthes de façon aléatoire. Implantez et testez vos algorithmes et répondez à certaines questions du type :

- (1) La génération est-elle uniforme (chaque labyrinthe apparaît avec la même probabilité) ?
- (2) Quelle est la complexité de votre algorithme ?
- (3) Á quoi "ressemblent" de très gros pseudo-labyrithes ou labyrinthes tirés aléatoirement ?

2.4. **D'autres directions...** Ce problème a été posé lors du tournoi des jeunes mathématiciennes et mathématiciens 2014. Vous trouverez sur le site du tournoi, <http://www.tfjm.org> d'autres pistes de recherche sur les labyrinthes.