

## TP - Labyrinth

### Description

Vous devez résoudre un problème de labyrinthe en utilisant python et le framework [DEAP](#) (implémentation de l'algorithme génétique).

Le labyrinthe est représenté par une matrice  $A$  de taille  $N \times N$ , où

- $a_{ij} = 0$  si la cellule est praticable
- $a_{ij} = 1$  si la cellule est un mur (non-praticable)

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

La cellule de départ est située en (0,0) la cellule d'arrivée est en bas à droite. Les déplacements autorisés sont les quatre points cardinaux nord, sud, est, ouest.

### Matériel

L'archive `data.zip` contient :

- Le module `labyrinth.py`
  - coeur logique de l'application
- Le notebook `notebook.ipynb`
  - terrain d'exécution et d'expérimentation
- Le fichier `grid10.npy`
  - sérialisation numpy d'une grille exemple de 10x10

Un début d'implémentation a été réalisé pour vous aider à comprendre la structure du TP.

### But

Implémenter la méthode `solve_labyrinth(...)` du module `labyrinth.py`. La signature de cette fonction **ne doit pas être modifiée**.

### Rendu

A rendre par mail à [hatem.ghorbel@he-arc.ch](mailto:hatem.ghorbel@he-arc.ch) ou sur cyberlear [stefano.carrino@he-arc.ch](mailto:stefano.carrino@he-arc.ch) avant le **dimanche 28 novembre 2021, 23h59**:

- Code : seul le fichier `labyrinth.py` **complété** est à rendre.

- Documentation : ajouter un fichier de texte (.md, .docx, .pdf, .txt) expliquant et justifiant **étape par étape** votre solution (encodage d'un chromosome, fonction de fitness, sélection, crossover, mutation, critère(s) d'arrêt et autres spécificités).

## Évaluation

Les critères suivants seront observés :

- Qualité du code et de la documentation
- Respect de l'algorithme génétique

Votre implémentation sera ensuite testée sur plusieurs grilles au travers de la méthode `solve_labyrinth(...)`.

Les conditions suivantes seront vérifiées :

- Est-ce que l'algorithme crash ?
- Est-ce que l'algorithme parvient à retourner une solution ? (après un temps max.)
- Est-ce que la solution est possible ?
- Est-ce que la solution est bonne ? (proximité avec la solution optimale)

Les grilles suivantes seront utilisées :

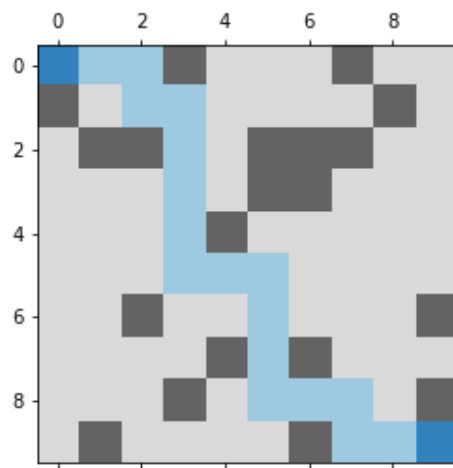
Taille	Temps max.
10x10	10s
15x15	15s
20x20	30s
30x30	60s
40x40	90s
30x30*	180s
30x30**	10s

\* grille non-aleatoire

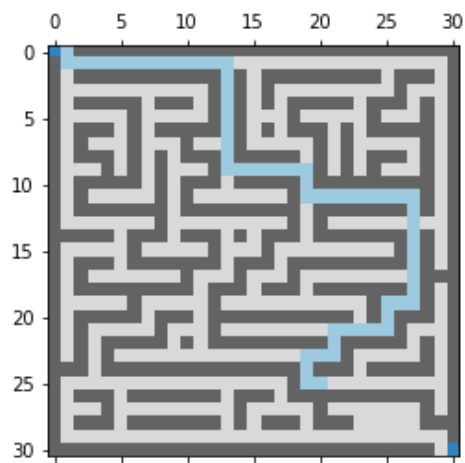
\*\* grille impossible

## Exemples

### Easy (10x10)



### Non aléatoire (30x30) – Le plus difficile !



## Remarques

- Travail individuel, noté

Bonne chance !