

# Construire un labyrinthe

Christophe Viroulaud

Terminale - NSI

**Algo 23**

1. Types de labyrinthe
2. Construire un labyrinthe
3. Utiliser une bibliothèque graphique

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

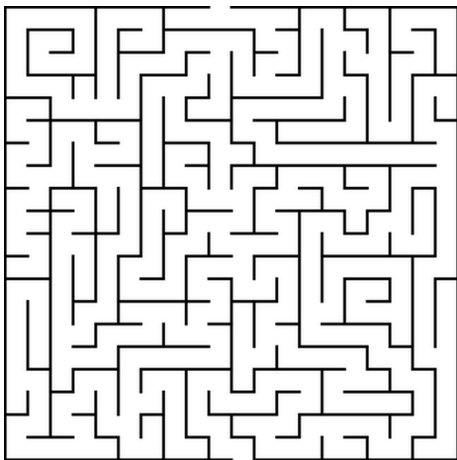


FIGURE 1 – Un labyrinthe peut être représenté par un graphe.

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

## Remarque

Il existe plusieurs catégories de labyrinthe. Considérons un labyrinthe où :

- ▶ tous les sommets sont atteignables,
- ▶ on ne peut pas tourner en rond.

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

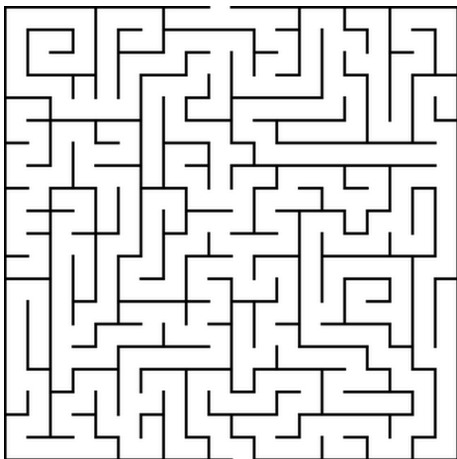


FIGURE 2 – Un labyrinthe est un graphe non orienté et acyclique.

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

1. Types de labyrinthe
2. Construire un labyrinthe
3. Utiliser une bibliothèque graphique

# Construire un labyrinthe

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

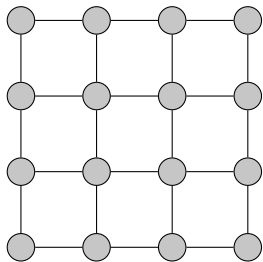


FIGURE 3 – Graphe de départ

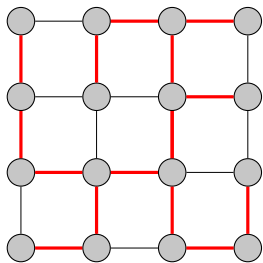


FIGURE 4 – En effectuant un parcours en profondeur, on peut construire un labyrinthe.



Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

1. Types de labyrinthe
2. Construire un labyrinthe
3. Utiliser une bibliothèque graphique

## À retenir

La bibliothèque `tkinter` est intégrée dans l'interpréteur Python. Elle permet de créer un affichage graphique.

```
1 import tkinter
2
3 TAILLE = 5
4 DIM = 50
5
6 fenetre = tkinter.Tk()
7 fenetre.title("Dessiner")
8 canevas = tkinter.Canvas(fenetre,
9                             width=DIM * TAILLE,
10                             height=DIM * TAILLE,
11                             bg="#FFFFFF")
12 canevas.pack()
```

Code 1 – Créer une fenêtre `tkinter` et un canevas.

```
1 canvas.create_line(0, 0, 10, 10, fill="black",  
    width=2)
```

Code 2 – Tracer une ligne de (0,0) à (10,10)

### Activité 1 :

1. Créer une grille de **TAILLE** lignes et **TAILLE** colonne.
2. Placer la ligne suivante en fin de programme :

```
1 fenetre.mainloop()
```

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

Types de  
labyrinthe

Construire un  
labyrinthe

Utiliser une  
bibliothèque  
graphique

```
1 for i in range(TAILLE):
2     canvas.create_line(i*DIM, 0,
3                         i*DIM, TAILLE*DIM,
4                         fill="black", width=2)
5     canvas.create_line(0, i*DIM,
6                         TAILLE*DIM, i*DIM,
7                         fill="black", width=2)
```

## Activité 2 :

1. Télécharger et extraire le dossier compressé `labyrinthe-annexe.zip` sur le site <https://cviroulaud.github.io>
2. Analyser le code. Comprendre le mécanisme pour diriger le cercle bleu.