

## Seminar 7 (Testat)

---

### Aufgabe 1: Backtracking

In der Vorlesung haben wir bereits kurz das Konzept des Backtrackings anhand eines Pseudocode-Beispiels besprochen. Im Folgenden soll ein lauffähiger Algorithmus in Python entwickelt werden, der den kürzesten Fluchtweg aus einem Labyrinth ermitteln kann. Gehen Sie dabei z.B. wie folgt vor:

- Übernehmen Sie die beiden Funktionen zum Einlesen eines Labyrinths und zur grafischen Ausgabe dieses Labyrinths aus der FloodFill-Aufgabe.
- Erstellen Sie eine Funktion `is_free(row_number, column_number)`, die bei Übergabe einer Zeilen- und einer Spaltennummer ermitteln kann, ob das zugehörige Feld gültig, d.h. ob es innerhalb des Spielfelds liegt, und frei ist, d.h. nicht mit einer Mauer belegt ist.
- Erstellen Sie eine Funktion `is_escape(row_number, column_number)`, die bei Übergabe einer Zeilen- und einer Spaltennummer ermitteln kann, ob das zugehörige Feld ein Ausgang aus dem Labyrinth ist. Ausgänge sind im Labyrinth mit "E" (Escape) markiert.
- Erstellen Sie die Funktion `find_escape(row_number, column_number, route=())`, der Zeilen- und einer Spaltennummer eines Feldes übergeben werden können, von dem aus die Suche nach einem Ausgang gestartet wird. Außerdem soll der Funktion die bisher gelaufene Route als Tupel von Koordinaten übergeben werden. Die Funktion soll, sofern das Feld ein Ausgang ist, die Koordinaten der Route zurückgeben. Beachten Sie bei der Implementierung:
  - Es sollen keine Felder betreten werden, die auf der aktuellen Route bereits betreten wurden, um Schleifen zu vermeiden.
  - Sofern mehrere Fluchtrouten gefunden wurden, soll nur die kürzeste berücksichtigt werden.

Zum Programmablauf:

- Einlesen der Datei `field2.txt`
- Starten der Suche bei den Koordinaten (1,1)
- Grafische Ausgabe des Fluchtwegs

Hier sind Sie.

Hier ist der  
Ausgang.

