

1. Labirentte En Kısa Yol Problemi

Bu ödevde, iki boyutlu bir labirentte başlangıç noktasından hedef noktaya giden en kısa yolu bulmanız gerekmektedir. Labirent, 0'lar ve 1'lerden oluşan bir matris şeklinde verilecektir:

- 0: Duvar (geçilemez)
- 1: Açık yol (geçilebilir)

Kullanabileceğiniz 4 hareket bulunmakta, bunlar:

- Yukarı
- Aşağı
- Sağ
- Sol

Çapraz hareket kullanılmayacak.

Genişlik Öncelikli Arama (BFS - Breadth First Search) algoritmasını kullanarak, başlangıç noktasından hedef noktasına ulaşan en kısa yolun uzunluğunu bulmanız istenmektedir. Eğer hedefe ulaşan bir yol yoksa çıktı olarak -1 değerini döndürmelisiniz.

Labirentin satır ve sütun değerleri 1 ile 100 arasında olacak. Girdi olan değerler:

`maze: List[List[int]], start: Tuple[int, int], end: Tuple[int, int]`

şeklinde verilecek. Çıktı olarak bir en kısa yol için bir int değeri ve takip edilen yolun list şeklinde verilmesi gerekiyor.

Örnek Labirent Değerleri:

Labirent:

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

Başlangıç: (0, 0)

Hedef: (3, 3)

Beklenen Çıktı:

```
$python3 cok_zor.py
```

```
[1, 0, 1, 1]
```

```
[1, 1, 1, 0]
```

```
[0, 1, 0, 1]
```

```
[1, 1, 1, 1]
```

```
Shortest distance: 7
```

```
Path taken: [(0, 0), (1, 0), (1, 1), (2, 1), (3, 1), (3, 2), (3, 3)]
```

Labirent

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

Başlangıç (0, 0)

Hedef (3, 2)

Beklenen Çıktı:

```
$python3 cok_zor.py
```

```
[1, 0, 0]
```

```
[0, 0, 0]
```

```
[0, 1, 0]
```

```
Shortest distance: -1
```

```
No path found.
```

Labirent

$$[1]$$

Başlangıç (0, 0)

Hedef (0, 0)

Beklenen Çıktı:

```
$python3 cok_zor.py
```

```
[1]
```

```
Shortest distance: 0
```

```
Path taken: [(0, 0)]
```

2. Çalışan kodunuzu GitHub'a yükleyip repositoryyi k.yavuzkurt1@gmail.com adresine gönderin.