



Veri Yapıları ve Algoritmalar Dönem Projesi

2021-2022 Bahar

Muhammet Kayra Bulut

20011901

Dersi Veren: Göksel Biricik

Video Linki: <https://www.youtube.com/watch?v=KY2pqpdIcDw>

Problem: Labirent Problemi

Problem tanımı: Basitçe verilen labirentte, verilen başlangıçtan yine verilen bitişe DFS (Depth First Search) yaklaşımıyla gidilen yolu matris formatında bulup, adım adım takip edilebilir şekilde görselleştirmemiz isteniyor.

Gerçekleştirdiğim çözümün tanımı: Verilen labirenti dosyadan okuduktan sonra, 'solveMaze' isimli fonksiyonum yardımıyla, labirentte gerçekten başlangıç ve bitiş var mı diye kontrol ettim. Eğer gerçekten bu ikisi de varsa 'solveMaze' isimli fonksiyonum kendi içinde 'solveMazeUtil' isimli fonksiyonu çağırıyor ve başlangıç indislerini de parametre olarak gönderiyor. 'solveMazeUtil' isimli fonksiyonsa, rekürsif bir fonksiyon ve 'delay' fonksiyonu yardımıyla 0.1 saniyede bir her çağırıldığında 'matrisPrint' fonksiyonu yardımıyla labirent matrisinin anlık durumunu yazdırıyor. Sonrasında bulunduğu konum eğer duvar değilse, sırasıyla 'isSafe' fonksiyonu kontrolünde "Aşağı-Sağ-Yukarı-Sol" komşuluk sırasıyla kendini bir daha çağırıyor ve bu süreçte kendi konumunu da matriste gezilmiş olarak işaretliyor. Aynı zamanda 'elma' ya da 'duvar' ile karşılaşılırsa gerekli puan güncellemelerini yapıyor. 'matrisPrint' fonksiyonuysa, görsel olarak rakamları anlamlandırdığım 'mySwitchCase' fonksiyonu yardımıyla ilgili görselleştirme işlemlerini yapıyor.

Örnek ekran çıktıları (Çok fazla olduğu için sadece 3 tane kullandım)

En başlarda

```
Puan: ---> 0
+ - + - + - + - + - + - + - +
| *b |   |   0 |   |   |   |   c -
+ * + - +   +   +   +   + - +   +
| *   |   |   |   |   |   |   |
+ * +   +   +   + - +   + - +   +
| * |   |   0 |   |   |   0 |   |
+   + - + - +   + - + - + - +   +
|   |   |   |   |   |   |   |
+ - +   +   + - + - +   + - +   +
|   |   |   |   |   |   |   |
+ - + - + - + - + - + - + - +
```

Ortalara doğru

```
Puan: ---> -45
+ - + - + - + - + - + - + - +
| *b |   |   0 |   |   |   |   c -
+ * + - +   +   +   +   + - +   +
| *   |   |   |   |   |   |   |
+ * +   +   +   + - +   + - +   +
| * |   |   0 |   |   |   0 |   |
+ * + - + - +   + - + - + - +   +
| * * *   |   |   |   |   |   |
+ - + * +   + - + - +   + - +   +
|   * * * * * * * * * * * * *
+ - + - + - + - + - + - + - +
```

En son kısım

```
+ - + - + - + - + - + - + - +
| *b |   0 |   |   |   |   *c -
+ * + - +   +   +   + - + * +
| *   |   |   |   |   * |
+ * +   +   + - +   + - + * +
| * |   0 |   |   0 | * |
+ * + - + - +   + - + - + * +
| * * *   |   |   * * * |
+ - + * +   + - + - + * + - +
|   * * * * * * * * *   |
+ - + - + - + - + - + - + - +
Yol bulundu... Son durumdaki puan ----> -65
```

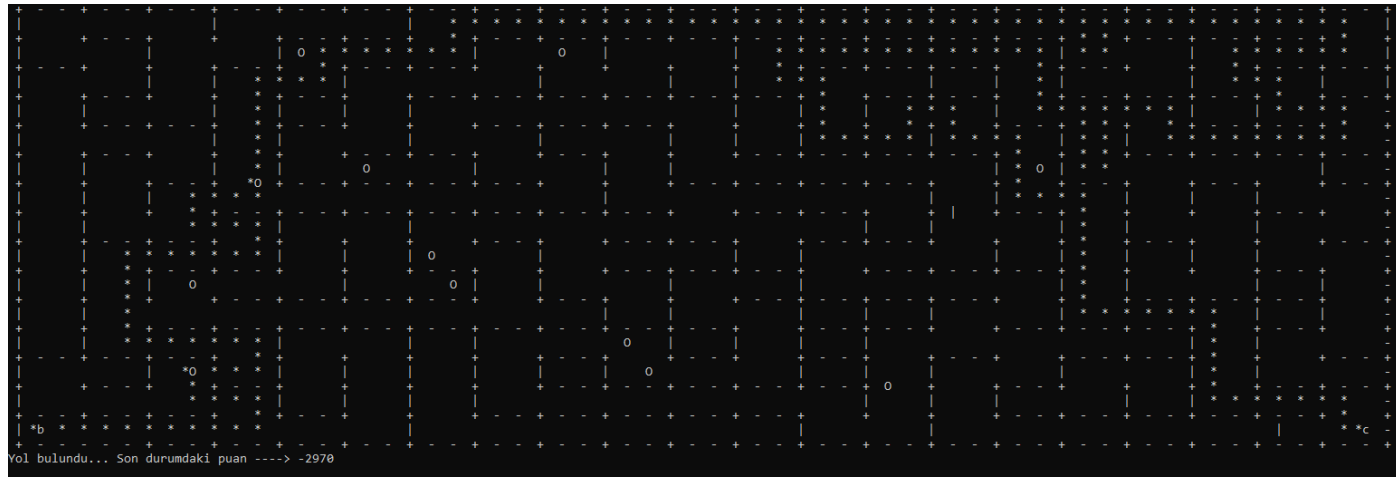
Program çalışırken txt dosyamın için

```
maze.txt - Not Defteri

Dosya  Düzenle  Görünüm

11 15
-3 1 -3 1 -3 1 -3 1 -3 1 -3 1 -3
1 -1 1 0 0 2 1 0 1 0 1 0 0 -2 1
-3 0 -3 1 -3 0 -3 0 -3 0 -3 1 -3 0 -3
1 0 0 0 1 0 0 0 1 0 0 0 0 0 1
-3 0 -3 0 -3 0 -3 1 -3 0 -3 1 -3 0 -3
1 0 1 0 0 2 1 0 0 0 0 2 1 0 1
-3 0 -3 1 -3 1 -3 0 -3 1 -3 1 -3 0 -3
1 0 0 0 0 0 0 0 1 0 0 0 0 0 1
-3 1 -3 0 -3 0 -3 1 -3 1 -3 0 -3 1 -3
1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
-3 1 -3 1 -3 1 -3 1 -3 1 -3 1 -3 1 -3
```

'Kullanılacak Labirent' isimli labirenti çalıştırdığımda aldığım sonuç çıktı:



Program çalışmadan önceki txt dosyamın içi:

[illegible]