

Yolunu Bul

Turan Öz

Bilişim Sistemleri Mühendisliği
(Teknoloji Fakültesi)
Kocaeli Üniversitesi
Kocaeli, Türkiye
181307006@kocaeli.edu.tr
turanoz@turanoz.com.tr

Abstract—In this assignment, an application that provides a recursive solution to the maze problem will be developed. Mazes will be input to the program in the form of simple text files. (txt extension) and the program output will be to find the correct path in the maze.

Keywords—maze solver, recursive, dictionary, graph, list

Özet—Bu ödevde, labirent sorununa özyinelemeli bir çözüm sağlayan uygulama geliştirilecektir. Labirentler programa basit metin dosyaları şeklinde girilecektir. (txt uzantısı) ve programın çıktısı labirentte doğru yolu bulmak olacaktır.

Anahtar Kelimeler—labirent çözücü, rekursif, sözlük, graf, liste

I. GİRİŞ

Kullanıcının W:Wall, P:Path, S:Start, F:Finish olacak şekilde metin dosyasına, ardışık, kare matris şeklinde oluşturduğu labirenti çözen, bu çözümü başka bir metin dosyasına aktarmasını amaçlayan uygulamadır. Birçok algoritma (BFS,DFS,A*,Dijkstra) ile çözülebilir. Benim yaptığım yöntemde kullanıcın girmiş olduğu metin dosyasını önce listeye aktaran daha sonra komşularını buldurarak sözlük içine ekleyen böylelikle ağırlıksız graf yapısı oluşturup, bu graf yapısını rekursif bir şekilde komşuluk üzerinden çözüme kavuşturan uygulama hazırladım.

II. KULLANIM KOLAYLIĞI

A. Daha Az Kod

Yazmış olduğum programda daha az kod satırı olmasına rağmen devamlı tekrar olduğu için bilgisayarı yoran bir algoritmaya sahip

B. Gidilebilecek Tüm Yollar

Deneyimlerime göre yazdığım algoritma gidilebilecek tüm yolları bir listeye aktarıp, en kısa yöntemi çıktı olarak hem metin dosyasına hem de dos ekrana yazdırıyor.

III. SÖZDE KOD

- Başla
- Girilen dosyayı oku ve liste içine ekle
- S,H,F noktalarının konumunu belirle
- S,H,F,P yollarının komşularını konum noktaları ile birlikte sözlüğe ekle
- Oluşan ağırlıksız graf yapısından S,H,F noktaları arasında komşuları üzerinden sonuca ulaşana kadar dolaş
- Bulunan tüm yolları listeye ekle
- En kısa yoldan hedefe ulaşan yol güzergahını return et
- Eğer H hücresi varsa başlangıç bitiş değerlerini uygun hale getir.

- Başlangıç : S, Bitiş: F, H:Güçlendirilmiş Hücre, Yollar :1, Duvarlar: 0, olacak şekilde ekrana ve metin dosyasına yazdır.
- Bitir

IV. EKRAN GÖRÜNTÜLERİ

```
guclugirdi.txt
1  WPPWWWWW
2  WWPWPSWW
3  WWPWPWWW
4  P PPPPWWW
5  PWPWPWWW
6  H P P P P P W
7  W W W W W W P W
8  W W F P P P P W
```

Metin dosyası içine W,P,S,F değerleri kullanılarak örnek 8x8 matris görünümü labirent eklenmiştir.

C:\Users\turan\Desktop\son>py main.py guclugirdi.txt cikti.txt

Terminal ekranından program çalıştırılmıştır.

Başlangıç Bitiş Güçlendirme Hücre Noktaları: [['S', '1,5'], ['H', '5,0'], ['F', '7,2']]

Çıktı olarak başlangıç bitiş güçlendirilmiş hücre koordinatları ekrana yazdırılmıştır.

```
ckiti.txt
1  0 0 0 0 0 0 0 0
2  0 0 0 0 1 S 0 0
3  0 0 0 0 1 0 0 0
4  1 1 1 1 1 0 0 0
5  1 0 0 0 0 0 0 0
6  H 1 1 1 1 1 1 0
7  0 0 0 0 0 0 1 0
8  0 0 F 1 1 1 1 0
```

Çıktı olarak metin dosyasına çözümünü yazdırmıştır.

KAYNAKÇA

- <https://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/>
- <https://www.udemy.com/course/veri-yapilari-egitim-serisi/>
- <https://www.pythontr.com/makale/python-sozlukler-ve-kullanimi-42>
- https://en.wikipedia.org/wiki/Maze-solving_algorithm