

TBL220 Python Programlama

Vize Proje Raporu

Bariş Adıgüzel - 181307059

ÖZET

Programa girdi olarak verilen metin dosyasından aldığımız belli kurallarla oluşturulmuş karakterleri bir labirente çevirip, ikinci bir metin dosyasına çözülmüş halini yazdırıyoruz.

Labirentin çözümünde kuracağımız algoritmayı alt problemlere ayırıp gerekli ilişkileri kurduğumuzda bir labirent oluşturup, o labirenti de çözümünü temelde ayrı bir listeye aktarıp programa ikinci girdi olarak verilen dosyanın içine yazdırıyoruz.

A. Giriş

Her şeyden önce problemimizin ne olduğunu anlamamız gerekmektedir. Projemizin problemi tanımlamak ilk aşamadır. Bizden istenenleri tanımladıktan sonra ikinci aşamamız problemin çözümü için gerekli unsurları belirlemektir. Kullanılacak verileri nasıl ve ne biçimde işleyeceğimizi belirleriz. Üçüncü aşamamız ise problemimizi çözecek algoritmayı kurmaktır. Nasıl bir yöntem izleyeceğimizi gerekirse bir yere not olarak algoritmayı kurarız. Dördüncü ve son aşamamız ise problemin çözümünü göstermektir.

Bu projede istenenler ilk olarak programa komut satırından girdi olarak verilecek .txt metin dosyalarından, verilen ilk girdinin içindeki verileri okumak ve onları bir labirente çevirmemiz istenmektedir. Metin dosyasının içi W,P,S,F karakterleri dışında hiçbir karakter içermemesi gerekmektedir. İkinci olarak dosyanın içindeki verileri okuduktan sonra bir labirent yani bir matrise çevirmemiz beklenmektedir. Bu kısımda iç içe list veri yapılarından yararlanılır. Bu kısımdan sonra problemimiz ikiye ayrılmaktadır. Klasik versiyon (Başlangıç noktasından bitiş çizgisine gitmek) ve Güç artırma hücresi (Bir güçlendirme hücresini bulmak).

Klasik versiyonumuzda bizden istenen S karakterinden F karakterine bir yol bulmaktır. Bunun için aslında daha önceden oluşturduğumuz matrise ihtiyacımız vardır. Gerekli hücreleri tarayıp S karakterinden F karakterine bir yol çizmek ve çözümünü de ikinci program girdisindeki dosyaya yazdırmak ile çözmüş oluruz.

Güç artırma hücresinde sınırlı bir güce sahibiz. Başlangıçta verilen güçle labirenti bitirmemiz imkansızdır. Labirenti bitirebilmek için güç hücresinden geçmemiz gerekmektedir. Güç hücremiz ise labirente H karakteri ile gösterilmektedir. H karakterini bulduktan sonra klasik yöntemdeki gibi F karakterini bulmaya çalışırız ve bulduktan sonra geçtiğimiz yolları da ikinci metin dosyasını yazdığımızda bu problemi de çözmüş oluruz.

B. Yöntem

Labirentleri çözmek için binbir çeşit algoritma mevcuttur. Benim kullandığım yöntem Recursive Backtracking algoritmasına benzetilmektedir. Kullandığım yöntemi kısaca özetlemem gerekirse:

- Başlangıç noktasından ilerlemeye başla
- Bulduğum hücre birden fazla yola gidebiliyorsa o hücreyi kayıt noktası olarak kaydet.
- Labirente çıkmaza gelinirse kayıt noktasının hangi yolundan ilerlenildiyse o hücreyi duvar yap
- Başlangıç noktasına dön ve tekrar başla

C. Pseudo Kod

1. Programa girdi olarak verilen girdileri al
2. İstenilen girdi sayısı girilmemişse kullanıcıyı bilgilendir ve 18. adıma git
3. İstenilen girdiler .txt dosya biçiminde değilse kullanıcıyı bilgilendir ve 18. adıma git
4. Girilen ilk girdi dosyası bulunamaz ise bir hata mesajı göster ve 18. adıma git. Fakat dosya mevcut ise 5. adıma git
5. Dosyanın içindeki verileri oku
6. Dosyadan alınan verileri bir labirente dönüştür
7. Başlangıç noktası, bitiş noktası ve eğer var ise güç hücresinin noktalarını belirle.
8. Oluşturulan labirent ile bir başlangıç noktası gönder
9. Eğer bir güç hücresi var ise bitiş noktasını duvar yap. Yok ise bitiş noktasını tekrar geri getir
10. Eğer güç hücresine gelindiye, hücreyi yol yap gezinti listesini yedekle ve ilgili güç hücresi adresini sil ve 8. adıma git
11. Mevcut noktamızın sadece gidiceği bir yol var ise ilerle ve gitmeden önce mevcut noktayı gezilmiş olarak kaydet ve 8. adıma git
12. Eğer mevcut noktamız gidiceği birden fazla yola sahip ise mevcut noktayı bir kayıt noktası olarak kaydet ve sırası ile gidilecek yollara ilerle ve 8. adıma git
13. Eğer gidilecek hiçbir yol yok ise ve bitiş noktasına ulaşılmadıysa en son ki kayıt noktasının neresinden ilerlenildiyse ilgili hücreyi duvar yap, gezilmiş olan yerleri sil ve 8. adıma git
14. Eğer bitiş noktasına gelindiye 15. adıma git.
15. Programda ikinci girdi olarak verilen dosyayı aç
16. Gezinti listesini kullanarak labirentteki ilgili değerleri, 1 veya 0 değerleri ile değiştir
17. Düzenlenmiş olan labirenti, 15. adımda açılan dosyaya yazdır ve kullanıcıya bilgilendir.
18. Programdan bitir.

Kaynakça

- [1] https://www.w3schools.com/python/python_file_handling.asp
- [2] <https://www.programiz.com/python-programming/recursion>
- [3] <https://www.programiz.com/python-programming/global-keyword>
- [4] <https://levelup.gitconnected.com/solve-a-maze-with-python-c9f0580979a1>

D. Örnek Sonuçlar

PPPSPPP	1, 1, 1, S, 0, 0, 0
PWWPWWP	1, 0, 0, 0, 0, 0, 0
PWWPWWP	1, 0, 0, 0, 0, 0, 0
PWPPWPP	1, 0, 0, 0, 0, 0, 0
PPWWWWP	1, 1, 0, 0, 0, 0, 0
WPPFWWP	0, 1, 1, F, 0, 0, 0

Solda bir klasik labirentin text formatındaki hali ve sağda ise labirentin çözümü

PPPSPPP	0, 0, 0, S, 0, 0, 0
PWWPWWP	0, 0, 0, 1, 0, 0, 0
PWWHWWP	0, 0, 0, 1, 0, 0, 0
PWWPWWP	0, 0, 0, 1, 0, 0, 0
PPWPWWP	0, 0, 0, 1, 0, 0, 0
WPPFWWP	0, 0, 0, F, 0, 0, 0

Solda bir güç artırma hücrelerine sahip labirentin text formatındaki hali ve sağda ise ilgili labirentin çözümü