

Problem Seti 8

Okumalar: 22.1-22.2 kısımlar, Bölüm (Chapter) 24.

Hem egzersizler hem de problemler çözülecek, ama sadece problemler teslim edilecektir. Egzersizler ders materyalini hazmettirmek amacıyla hazırlanmıştır. Her ne kadar egzersiz çözümlerini teslim etmeyeceksiniz de egzersizlerdeki konulardan sorumlu olacaksınız.

Sık sık bir problem için “bir algoritma bulun” isteğiyle karşılaşacaksınız. Bu konudaki yanıtınız kısa bir makale şeklinde olmalıdır. Konu paragrafı, çözdüğünüz problem ve sonuçlarınızı özetleyecek şekilde düzenlenmelidir. Makalenizin ana yapısında aşağıdaki bilgiler verilmelidir:

1. Algoritmanın İngilizce açıklaması ve eğer faydalı olacaksa Sözde kodu..
2. Algoritmanızın nasıl çalıştığını gösteren en az bir işlenmiş örnek veya şekil.
3. Algoritmanın doğruluğunun kanıtı (veya göstergesi).
4. Algoritmanın koşma süresinin çözümlenmesi.

Amacınız iletişim kurmaktır. Tam not sadece iyi açıklanan doğru yanıtlara verilecektir; ya da net olmayan açıklamalar daha düşük notlandırılacaktır.

Egzersiz 8-1. Kitaptaki 22.2-6 nolu egzersiz (539. Sayfa)

Egzersiz 8-2. Kitaptaki 24.1-3 nolu egzersiz (591. Sayfa)

Egzersiz 8-3. Kitaptaki 24.2-4 nolu egzersiz (595. Sayfa)

Egzersiz 8-4. Kitaptaki 24.3-2 nolu egzersiz (600. Sayfa)

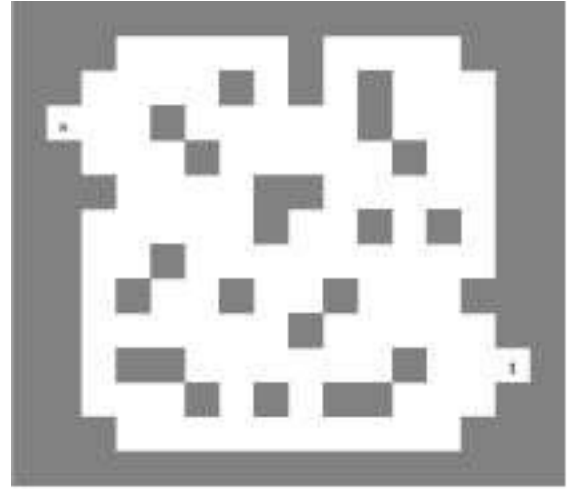
Egzersiz 8-5. Kitaptaki 24.5-2 nolu egzersiz (613. Sayfa)

Egzersiz 8-6. Kitaptaki 24.5-7 nolu egzersiz (614. Sayfa)

Problem 8-1. Sola Dönüş yok

Yeni bir Nexus Nano çaldınız, ancak arabanın sahibi, arabayı, The Spade denilen bir mekanik kilitle sola dönmelerini engelleyecek şekilde kitlemiş. Kaza yapmamak ve dikkat çekmemek için herhangi bir U dönüş de yapmamanız gerekiyor. Düz giderken hızlı gidebildiğinizden, sağa dönüşleri en aza indirerek, olabildiğince çabuk şekilde, aracın parçalarını satacağınız garajınıza ulaşmanız gerekiyor. Ancak eğer iki farklı yoldaki sağa dönüş sayısı aynı ise, düz gittiğiniz mesafeyi de azaltmak istiyorsunuz.

İyi ki elinizde şehrin haritası var, böylece sola dönmeden ve U dönüşü yapmadan bir yol planlayabilirsiniz. Daha da iyisi, haritanın $m \times n$ hücreli elektronik sürümü de mevcut. Bu hücreler ya boş(geçilebilir) ya dolu(geçilemez) şeklindedir. Her bir hücrede geldiğiniz yönde devam edebilir ya da sağa dönebilirsiniz. Sola dönmeye ihtiyacınız olmayan 2 örnek harita Şekil 1'de gösterilmiştir.



Şekil 1: 2 tane sola dönüşsüz harita. s'den t'ye olabildiğince az sağa dönüşlü, sola dönüşsüz ve U dönüşsüz yol bulun. Bu haritalar ile www.clickmazes.com/noleft/ixnoleft.htm adresindeki Java uygulamaları ile oynayabilirsiniz.

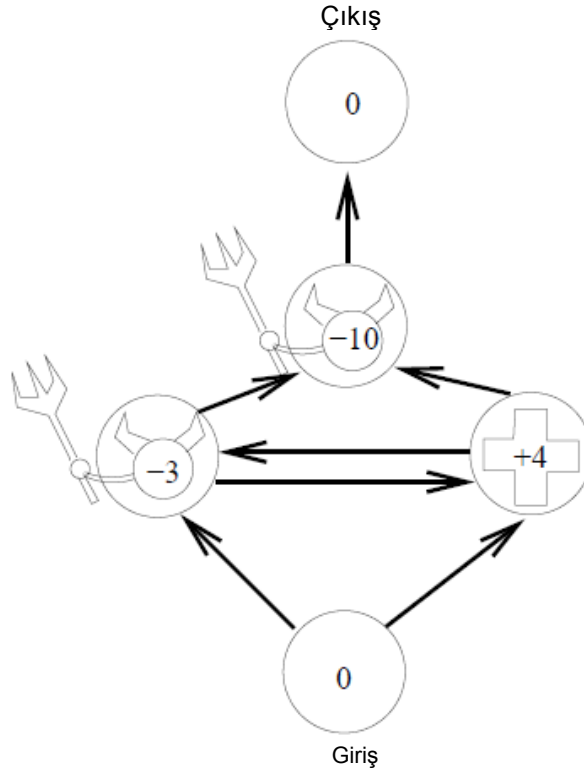
(a) $m \times n$ büyüklüğünde bir haritada, sola dönüşsüz, olabildiğince az sağa dönüşlü bir yol bulmak, veya böyle bir yol olmadığını belirlemek için verimli bir algoritma bulun. Algoritmanız aynı zamanda, eşit sağa dönüşlü yollar arasında, düz yol uzunluğunu da en aza indirmelidir. (*İpucu:* Soruyla ilgili bir sezgi edinebilmek için sol taraftaki harita için soruyu çözün.)

(b) a bölümündeki algoritmanız ile üretilen yol haritası çıktıktan sonra, polislin sizi takip ettiği durumda garajınıza kadar götürebileceğinizi düşünün. Takipten kurtulmak için en iyi stratejinin hızlı bir şekilde 2 sağ dönüş yapmak olduğunu düşünün. (örnek, komşu karelerde, 90 derecelik dönmeler 1 birim düz gitme ile kesilir)

Yani 2 tane ardışık sağa dönüş içeren ve hala sol dönmediğiniz, mümkün olduğunca az sağ dönüşe sahip, eşit sağa dönüşlü yollarda da en kısa düz yolu tercih edeceğiniz bir yol bulmak istiyorsunuz. Böyle bir yolu bulmak veya yol olmadığını belirlemek için verimli bir algoritma yazın.

Problem 8-2. Video Oyun Tasarımı

Profesör Cloud, bir senedir, uzun zamandır beklenen Take-home Fantasy XII isimli oyunun tasarımına danışmanlık yapıyor. Oyundaki seviyelerden birinde, oyuncular girişten çıkışa gitmek için bir labirentteki birçok odayı gezmek zorunda. Odalarda hayat iksiri veya canavar olabileceği gibi, odalar boş da olabilir. Oyuncu odalarda dolaşırken L yaşam puanları azalır veya artar. Hayat iksiri içmek L'yi artırırken, canavarla karşılaşmak L'yi azaltır. Eğer L 0'ın altına düşerse, oyuncu ölür.



Şekil 2 1-girilebilir labirent örneği.

Şekil 2'de gösterildiği gibi bu labirent, köşelerin odaları, tek yönlü kenarların da koridorları simgelediği bir $G=(V,E)$ yönlendirilmiş grafiği ile simgelenebilir. $f : V \rightarrow Z$ köşe ağırlık fonksiyonu odanın içindekileri simgeler.

- Eğer $f(v) = 0$ ise oda boştur.
- Eğer $f(v) > 0$ ise odada hayat iksiri vardır. Oyuncu odaya her girdiğinde L yaşam puanı $f(v)$ kadar artar.
- Eğer $f(v) < 0$ ise odada canavar vardır. Oyuncu odaya her girdiğinde L yaşam puanı $|f(v)|$ kadar azalır. L eksi olursa oyuncu ölür.

Labirente giriş odası $s \in V$, çıkış odası da $t \in V$ 'dir. s 'den her v 'ye ve her v 'den t 'ye bir yol vardır. Oyuncu, s girişinden $L = L_0$ yaşam puanı ile girişten oyuna başlar.

Basit olsun diye girişin boş olduğunu düşünün: $f(s)=0$

Profesör Cloud canavarları ve hayat iksirlerini labirente rastgele yerleřtirmek için bir program tasarlıyor, ama bazı labirentlerde girişten çıkışa gitmek $L_0 > 0$ olmak zorunda olduğundan mümkün olmayabiliyor. s 'den t 'ye giden yolda eğer oyuncu sağ kalabiliyorsa bu yol girilebilirdir.

Eğer $L_0=r$ ile başlayan bir labirentte girilebilir bir yol varsa, r -girilebilir labirenti tanımlayın.

Profesöre r 'nin en küçük değerini tanımlamak için verimli bir algoritma tasarlamasında yardımcı olun. Öyle ki, bu algoritma, verilen labirentin r -girilebilir olduğunu veya böyle bir r olmadığını belirlesin. (Gidiş yolundan puan almak için, bir labirentin verilen bir r için r -girilebilir olup olmadığı problemini çözün.)

(a) Şekil 2'deki labirent için güvenli bir yol bulun.

(b) Ağırlıkların kenarlarda olduğunu düşünerek soruyu denk bir probleme çevirin, ve bu denkliği kanıtlayın.

(c) Bu problem için hiç bir hayat puanı arttıran devre olmadığını varsayın. Verilen bir r için, labirentin r -girilebilir olup olmadığını nasıl kontrol edersiniz?

(d) Şimdi, hayat puanlarını arttıran bir devre olabileceğini varsayın. Verilen bir r için, labirentin r -girilebilir olup olmadığını nasıl kontrol edersiniz?

(e) r -girilebilir bir labirentte, en küçük r değerini nasıl bulabilirsiniz?