

ALGORİTMA TASARIMI ve ANALİZİ

2020 – 2021 Güz Dönemi
Kısa Sınav 2

Ödev Duyurulma Tarihi : 30 Aralık 2020 Çarşamba

Ödev Teslim Tarihi : 8 Ocak 2021 Cuma, Saat 23:59

1. TANIMLAMA

Bu ödevde yeraltı kanallarında yaşayan yavru Ninja kaplumbağalar ve Usta Splinter'in olduğu bir pizza toplama oyunu tasarlanmıştır. Bu oyunda Splinter Usta'nın yavru Ninja kaplumbağalar için yiyecek aramasını simüle eden C++ kodunu yazmanız beklenmektedir. Splinter Usta pizza ararken aşağıda belirtilen kurallara göre yeraltı kanallarında hareket etmektedir. Kodu yazarken Dijkstra Algoritmasını ve Knapsack Problem (1/0 Knapsack) algoritmasını kullanınız. Ayrıca ihtiyaç dâhilinde ilgili veri yapılarını da kullanabilirsiniz.



Şekil 1- Yeraltı Kanal Sistemi

Splinter Usta, 4 yavru Ninja kaplumbağayı (Rafael, Donatello, Leonardo, Michelangelo) beslemektedir. Her bir yavru Ninja'nın hayatta kalabilmesi için, zamanla azalan enerjileri bitmeden önce pizza yiyerek enerji alması gerekmektedir. Splinter Usta yavru Ninjalarını sağlıklı bir şekilde büyütmek için devamlı olarak pizza aramaktadır. Ayrıca Splinter Usta'nın kendisinin de gün içerisinde alması gereken belirli miktarda kalori bulunmaktadır. Splinter Usta her gün Ninja evinden başlayarak yeraltı kanallarında

dolaşmaktadır. Bu kanalları dolaşırken belirli kurallara göre hareket etmektedir. Ödevde Splinter Usta'nın optimum şekilde hareket etmesini sağlayacak gerekli algoritmaları yazmanız beklenmektedir.

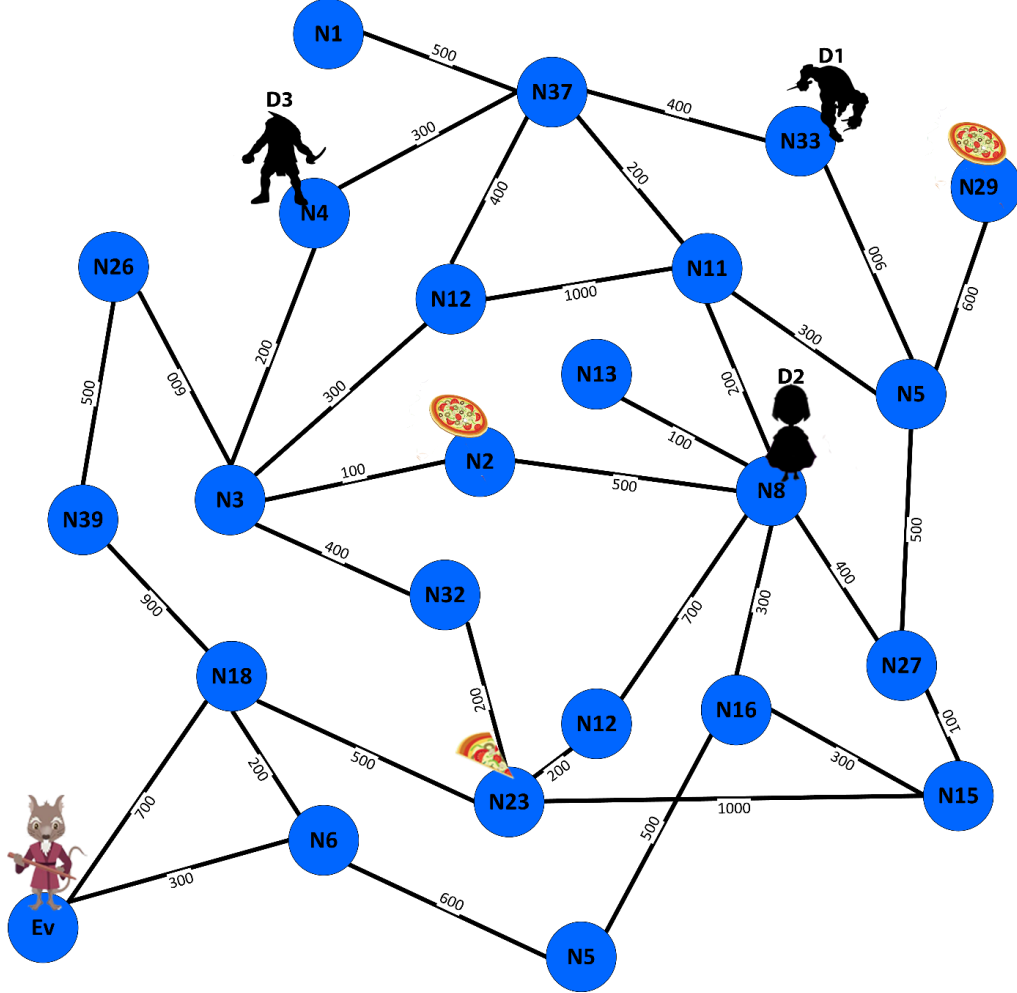
- Yeraltı haritasında kanal bağlantıları graf üzerinde Şekil-2'de gösterilmiştir.
- Splinter Usta haritada gezerken verilen düğümler (noktalar) arasında hareket edecektir. Maksimum miktarda kaloriye sahip pizzaları toplayıp Ninja evine dönecektir.
- Splinter ustanın çantasında taşıyabileceği pizza miktarı çantanın kapasitesine bağlıdır. En fazla çantanın hacmi kadar pizza taşınabilecektir.
- Tablo-1'de yeraltı kanal sistemine ait haritada düğümler ve düğümler arasındaki uzaklıklar verilmiştir. Splinter Usta kanal içerisinde gezinirken belirli miktarda kalori harcamaktadır. İhtiyaç duyduğu kaloriyi de bulduğu pizzalardan karşılamaktadır. Oyunda oluşturulan pizzaların kalori değerleri farklılık gösterebilmektedir. Pizzaların özellikleri (kalori değeri, hacmi...) tablo 2'te gösterilmiştir.
- Splinter usta keskin koku alma duyusu ile haritada bulunan bütün pizzaların konumunu bilmektedir. Ayrıca yavru Ninjaların enerji seviyesi 20 birimin altına düştüğünde yavru Ninjaların zor durumda olduğunu hisseder (bakınız *Uyari()* fonksiyonu).

2. PROBLEM

1. Yeraltı kanalları haritasında kanal bağlantıları arasındaki mesafeler *KanalBaglantiBilgileri.txt* dosyasında verilmiştir (bakınız Tablo-1).
2. Ninja evi ve diğer noktalar birer "düğüm" (bağlantı noktası) olarak ele alınacaktır (bakınız Şekil-2).
3. Her bir kaloninin bir birim enerji miktarına eşit olduğu varsayılacaktır.
4. Evden çıktığı anda Splinter Usta'nın 500 birim enerjisi, yavru Ninjaların ise her birinin 100 birim enerjisi vardır.
5. Splinter Ustanın depolayabileceği maksimum kalori miktarı 2000 kalori, yavru Ninjaların ise 300 kaloridir.
6. Splinter Ustanın hızı 2 m/sn dir.
7. Splinter Ustanın her 1000 metre mesafe için 200 birim enerji tüketmesi gerekmektedir.
8. Yavru Ninjaların her biri bir saatte 50 kalori enerji tüketmektedir.
9. Splinter Usta haritadaki hangi konuma gideceğini; pizzaların kalori, hacim ve bu pizzalara ulaşmak için harcaması gereken enerji miktarına göre karar verecektir.
10. Splinter Usta çantasında toplamda 2500 cm³ hacminde pizza taşıyabilir.
11. Splinter Usta pizza toplamaya giderken bazı noktalarda düşman saldırıları ile karşılaşabilir. Ancak düşmanların yerlerini bilmemektedir. Karşılaştığı düşmanı yenip yoluna devam edebilmesi için düşmanın direnç değeri kadar enerji harcaması ve düşmanın kaybettireceği süre miktarı kadar zaman harcaması gerekmektedir. Splinter ustanın düşman ile mücadelesinde geçen sürede sadece yavru Ninjaların zamana bağlı olarak enerjileri azalmaya devam eder. Splinter ustanın düşmanı yenmesi için harcaması gereken enerji miktarı ve geçen süre (dk) *DusmanBilgisi.txt* dosyasında verilmiştir.
12. Splinter Usta pizza toplamak için optimal yolu kullanarak kanal içerisinde gezecektir. Gezeceği noktaları, minimum kalori tüketecek ve maksimum kalori toplayacak şekilde belirleyecektir.
13. Program başlatıldığında kullanıcıdan aşağıdaki bilgiler alınacaktır.
 - Harita üzerine yerleştirilecek pizza sayısı
 - Pizzaların yerleştirileceği nokta(ID) bilgileri
 - Pizza ID(PizzaBilgisi.txt dosyasından seçilecek pizza)

14. Splinter Ustanın harita üzerinde pizzaları toplayıp eve ulaştığı anda hareketlerin özetini gösterecek **raporVer()** fonksiyon çalıştırılacak ve aşağıdaki bilgiler ekrana yazdırılacaktır.

- Yavru Ninjaların kalan enerji bilgileri
- Splinter Ustanın kalan enerji bilgisi, yolda harcadığı enerji miktarı ve eğer yolda enerjisi bitip de pizza yediyse yenilen pizza türü
- Kullanılan optimal yol bilgisi
- Karşılaşılan düşman bilgileri
- Toplamış olduğu pizza miktarı ve eve getirdiği pizza miktarı



Şekil 2- Kanal Sistemi Bağlantı Grafi

3. OLUŞTURULACAK FONKSİYONLAR

Main.cpp

grafOlustur() : KanalBaglantiBilgileri.txt dosyasından okunan bilgilere göre noktalar arasındaki bağlantı ve uzaklık bilgilerini tutan graf oluşturulacaktır.

pizzaOlustur(): Kullanıcıdan harita üzerinde hangi noktalara pizza ekleneceği bilgisi istenecektir. Bunun için kullanıcıdan öncelikle kaç adet pizza ekleyeceği bilgisi alınmalıdır. Daha sonra kullanıcıdan;

pizzaların kanalda konulacak ilgili noktaların ID si alınacaktır. Ayrıca belirlenen noktalara eklenecek pizza türünü ifade eden PizzaID bilgileri alınacaktır. Bu şekilde kanal içerisinde belirli noktalara istenilen türde pizza eklenmiş olacaktır.

Uyari(): Usta Splinter'ın ve Yavru Ninjaların kalori değerleri belirli değerin altına düştüğünde ekrana uyarı bilgisi verilecektir (bakınız yapılacaklar/madde 4).

enerjiMiktariHesapla(): Splinter usta kanal içerisinde bir düğümde başka bir düğüme geçtiğinde kalan enerji seviyesi ve yavru Ninjaların kalan enerji seviyesi hesaplanacaktır.

optimalYolBul() : Splinter Usta'nın kanal içerisinde pizzaları toplayıp Ninja evine dönmesi için en uygun yollar belirlenir ve Splinter Usta bu yollara (güzergâh bilgisi) göre harekete başlar.

raporVer(): Splinter Ustanın harita üzerinde pizzaları alıp eve ulaştığı anda görevi tamamlamış olarak kabul edilecektir. Hareketlerin özetini göstermek için aşağıdaki bilgiler ekrana yazdırılacaktır (bakınız şekil 3).

- Yavru Ninjaların kalan kalori bilgileri
- Splinter Ustanın kalan kalori bilgisi, yolda harcadığı kalori miktarı ve eğer yolda enerjisi bitip de pizza yediyse yenilen pizza ID'si
- Kullanılan optimal yol bilgisi
- Karşılaşılan düşman bilgileri
- Toplamış olduğu pizza miktarı ve eve getirdiği pizza miktarı

Splinter Ustanın Uğradığı Optimal Yollar:	EV->N20->N19-> N18(Pizza)->N1-> N9(Dusman)-> N8-> N14-> N23(Pizza)->N24->N20-> EV
Karşılaşılan düşman bilgileri:	D3, D6
Toplamış olduğu pizzaların kalorisi:	5200
Eve getirdiği pizzaların kalorisi:	1800
Usta Splinter'ın kalan enerji seviyesi:	65
Usta Splinter'a düşen pizza kalori miktarı:	1000
Yavru Ninjaların kalan enerji seviyesi:	10
Yavru Ninjalara düşen pizza kalori miktarı:	1000/4=250

Şekil 3– Rapor Fonksiyon Çalıştırıldığında Ekrana Yazdırılan Bilgiler

4. GİRDİ VERİ DOSYALARI

Düğüm ID	Ninja evi	N1	N2	N3	N4	N5	...	N40
Ninja evi	0	-	-	-	1000	-	...	2000
N1	-	0	-	-	500	700	...	-
N2	-	-	0	-	-	500	...	-
N3	-	-	-	0	700	-	...	5000
N4	1000	500	-	700	0	600	...	3500
N5	-	700	500	-	600	0
...
N40	2000	-	-	5000	3500	-	-	0

Tablo 1- Haritadaki düğümler ve aralarındaki uzaklıklar

Pizza ID	Pizza Boyutu	Kalorisi(Cal)	Hacim(cm ³)
P1	Büyük	1000	1000
P2	Orta boy	500	600
P3	Küçük boy	300	400
P4	Dilim pizza	100	200

Tablo 2- Pizza Özellikleri

Düşman ID	Direnç (Kalori)	Konumu	Kaybettirdiği Süre (dakika)
D1	60	N4	10
D2	80	N2	5
D3	100	N5	10
D4	50	N7	5
D5	300	N11	30
D6	200	N15	20

Tablo 3- Düşman bilgileri

5. YAPILACAKLAR

1. *KanalBaglantiBilgileri.txt* dosyasından mesafe bilgileri okunacak ve **grafOlustur()** fonksiyonu ile okunan bilgiler kullanılarak bağlantı grafi oluşturulacaktır.
2. Splinter Usta oluşturulan graf üzerinden **optimalYolBul** fonksiyonunu kullanarak minimum kalori harcayacak, çantasına maksimum kaloride pizza yerleştirecek şekilde optimal yol güzergahı oluşturacaktır.
3. Pizza bilgileri *PizzaBilgileri.txt* dosyasından okunacaktır. Bu bilgilere göre harita üzerinde oluşturulacak olan pizzaların türünü kullanıcı seçecektir. Öncelikle kullanıcıdan haritaya kaç adet pizza oluşturulacağı bilgisi alınır. Daha sonra kullanıcıdan kanalda hangi bağlantı noktasında(DüğümID) ve hangi türde pizza(PizzaID) oluşturulacağı bilgisi alınacaktır.
4. Oyun başlatıldıktan sonra Splinter Ustanın enerji miktarı 200 birime düştüğünde veya yavru Ninjaların enerji miktarı 20 birime düştüğünde **Uyari** fonksiyonu çağrılır. Yavru Ninjaların ve Usta Splinter'in kalan enerji miktarları ve Usta Splinter'in eve ulaşması için takip etmesi gereken kalan yol bilgileri Şekil-3'te belirtildiği gibi ekrana yazdırılmalıdır. Aşağıdaki kıstaslar dikkate alınacaktır;
 - a. Uyarı fonksiyonu çalıştığı anda Usta Splinter'in enerji miktarı hedefe ulaşabilmesi için yeterli değilse ve çantasında pizza varsa eve ulaşabilecek minimum kaloriye sahip pizza yenilecektir.
 - b. Yavru Ninjaların enerji seviyesi 20'nin altına düşerse, Usta Splinter kendisinin de enerjisini karşılayacak şekilde yavrulara acil olarak en kısa yoldan pizza götürmeye çalışacaktır.
 - i. Eğer Splinter ustanın çantasında kendisine ve yavrulara yetebilecek yeterli pizza var ise doğrudan en kısa yol ile eve dönecektir.
 - ii. Çantada pizza yok ise ulaşabileceği en kısa mesafedeki yeterli miktarda pizzayı veya pizzaları alıp eve gidecektir.
 - iii. Pizzaların Yeterli olup olmadığına; kalan enerji seviyesi ve eve gidene kadar harcanacak olan enerji miktarının farkından elde edilen net enerji seviyesine göre karar verilecektir.
 - c. Yavruların enerji miktarı Usta Splinter'in eve ulaşabileceği vakte kadar yeterli olmaz ise Usta Splinter hızını 2 katına çıkaracak bu durumda ise Usta Splinter'in tükettiği enerji miktarı 2 katına çıkarılacaktır.
 - d. Bu kıstaslar uygulandığı halde yavru Ninjaların veya Usta Splinter'in enerjisi tükenirse program sonlandırılacak, **raporVer()** fonksiyonu çalıştırılacak ve ekrana "Görev Başarısız" mesajı yazdırılacaktır.
5. Usta Splinterin graf üzerinde pizza aramak için takip edeceği optimal yol oluşturulurken düşman bilgileri hesaba katılmayacaktır. Optimal yol oluşturulduktan sonra *DusmanBilgileri.txt* dosyasından okunan düşman bilgileri dikkate alınacak ve graf üzerindeki noktalara eklenecektir.
6. Splinter Usta düşmanın bulunduğu noktalardan geçerken o düşmanın direnci kadar enerji kaybedecektir. Düşmanın kaybettirdiği süre kadar yavruların enerjisinde azalma olacaktır.
7. Splinter usta eve döndüğünde çantadaki pizzaların yarısını kendisine geri kalanını da 4 e bölerek yavru Ninjalara verecektir.
8. Splinter Usta görevi tamamlayıp Ninja evine ulaştığı anda **raporVer** çalıştırılacak ve ekrana "Görev Tamamlandı" mesajı yazdırılacaktır.

6. DİKKAT EDİLMESİ GEREKENLER

1. Ödevleriniz değerlendirilirken elimizdeki farklı test dosyaları (**DusmanBilgisi.txt, KanalBaglantiBilgileri.txt, PizzaBilgileri.txt**) kullanılacaktır. Bu test dosyaları sizinle paylaştığımız örnek “*.txt” dosyaları ile aynı formatta olacaktır. Bu yüzden kodunuzun farklı boyutlarda veri alıp veri yazma işlemine izin verdiğinden emin olunuz.

2. Ödev C++ programlama diliyle yazılacaktır (C++ dili, C dilini desteklediğinden kodunuzun çalışıyor olması C++’a özgü hazırlanmış anlamına gelmez. C++ dilinde hazır **STL kütüphanelerini** kullanmayınız. Kullandığınız takdirde ödev değerlendirmeye alınmayacaktır. C++ diline özgü olmasına dikkat edin, nesne kullanımı, bellekte yer ayırma gibi işlemlerde dikkatli olun, malloc yerine new, free yerine delete gibi).

3. Ödevi hazırlarken doğru veri yapısını ve doğru algoritmayı kullandığınızdan emin olun!

4. Ödevlerinizi **algoritmataasarimodevi@gmail.com** adresine aşağıdaki formatta gönderiniz!

Örneğin, ödevi gönderen kişi Yiğit Yıldırım olsun ve öğrenci numarası 200500020 olsun.

Öncelikle, ödevde main dosya ismi **200500044_YigitYildirim_odev2.cpp** olacak. Ödevinizin bulunduğu klasörünüzün ismi de;

200500044_YigitYildirim_odev2.zip

şeklinde olacaktır. Yani göndereceğiniz ödev dosyasını, kendi öğrenci numaranız ve ad soyadınıza göre örnekte gösterildiği gibi adlandırın (dosyayı isimlendirirken Türkçe karakter kullanmayınız, boşluk bırakmayınız, ayırmayı alt çizgiler kullanarak yapınız).

5. Gönderdiğiniz e-postalarınızda karışıklık olmaması adına, ödevinizi gönderirken posta başlığı

(e-mail başlığı): **Algoritma Analizi ve Tasarımı Ödev-2** şeklinde olmasına dikkat ediniz.

6. Ödevlerinizi en geç teslim tarihi saatine (ilk sayfada) kadar göndermiş olduğunuzdan emin olunuz.

Ödev teslim tarih saatinden sonra gönderilen ödevler değerlendirilmeye alınmayacaktır!

7. Bu ödevin, genel dönem puanına etkisi **20 puandır**. Kopya ödev, İnternette hazır kopyaya yapıştır ödev teslim etmeyiniz. Kopya yazılımlarında benzerlik olacağından kendiniz yazınız. Bu tür durumların tespitinde öğrencinin **disiplin kuruluna sevk**i yapılacaktır. O yüzden ödevinizi kendiniz yapınız!

8. Göndereceğiniz kodun Linux konsol (terminal) ortamında çalışıyor olduğuna dikkat ediniz.

9. Ödevi Gönderirken sadece “*.cpp”, “*.h” dosyalarınızı gönderiniz. Ödevi gönderirken proje halinde göndermeyiniz. Aksi Takdirde ödev değerlendirmeye alınmayacaktır. Ayrıca ödevi gönderirken lütfen “*.exe” dosyası göndermeyiniz.