

2018-2019 Bahar Yarıyılı

Algoritma Analizi

Ödev – 1

Ders Yürütücüsü: Dr. Öğr. Üyesi M. Amaç GÜVENSAN

Konu : Rekürsif Algoritmalar

Problem: Gezgin Çekirge Oyunu

İki çekirge bir yolun başından sonuna doğru zıplayarak gitme oyunu oynamaktadırlar. Oyunun kuralları şöyledir :

- 1.) Çekirgelerden biri yol boyunca yan yana dizili taşlara sayılar yazar. Her sayı o taştan itibaren sağa veya sola kaçınıcı taşa zıplanabileceğini göstermektedir. Son taşın üstünde oyunun bittiğini göstermek için **her zaman 0** (sıfır) yazar.
- 2.) Diğer çekirge başlangıç taşından başlayarak her taşın üstünde yazan sayı kadar sağ veya sol yöne doğru zıplayarak son taşa gitmeye çalışır.
- 3.) Eğer herhangi bir yönde **yeteri kadar taş yoksa** çekirge o yöne doğru zıplayamaz. Örneğin 2. taşın üzerinde 3 yazıyorsa, bu taşın solunda sadece 1 taş olduğu için sola doğru zıplaması mümkün değildir.
- 4.) Çekirge son taşın üzerine geldiğinde oyun biter. Son taşa gitmenin **birden fazla yolu** olabilir.

Bu ödevde her birinin üzerinde kaç taş zıplanacağı yazılı **n** adet taş değerini kullanarak (**N.** taşın değeri her zaman 0 olmalıdır) gezgin çekirgenin **başlangıç taşından son taşa gidip gidemeyeceğini bulan rekürsif fonksiyonu** ve bu fonksiyonu kullanan ana programı yazınız.

Rekürsif fonksiyon : int solvable(int start, int leapCount[], int n)

Start : hareket başlangıç noktası

LeapCount[] : her taştaki sıçrama sayılarının saklandığı dizi

n : dizi uzunluğu

Dönüş değeri : Çekirge son taşa gidebilirse : 1, gidemezse : 0

Örnek 1 :

Taşlara aşağıdaki sayılar yazılmış olsun. Oyuncu çekirge başlangıç taşı (boyalı taş) üzerindedir.

Başlangıç :

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

Bu taşın solunda taş olmadığı için sağa doğru 3. taşa zıplamak zorundadır. Zıplayarak sağdan 4. taşa (boyalı taş) gelir.

1. zıplama : 3 taş sağa

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

Bu taştan sağa veya sola doğru 1 taş zıplayabilir. Sola doğru zıpladığını varsayarsak yeni yeri aşağıdaki gibi olur.

2. zıplama : 1 taş sola

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

Çekirge oyunu aşağıdaki zıplamalarla bitirebilir.

3. zıplama : 4 taş sağa

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

4. zıplama : 2 taş sağa

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

5. zıplama 3 taş sola

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

6. zıplama : 4 taş sağa

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

Örnek 2 :

Oyunun bitememesi de mümkündür. Örneğin aşağıdaki durumda çekirge sürekli sağa ve sola doğru 3 taş ileriye zıplamak zorundadır. Son taşa hiçbir zaman ulaşamaz.

Başlangıç:

3	1	2	3	0
---	---	---	---	---

1. zıplama:

3	1	2	3	0
---	---	---	---	---

2. zıplama

3	1	2	3	0
---	---	---	---	---

.....

k. zıplama

3	1	2	3	0
---	---	---	---	---

Ödev Teslimi ile ilgili önemli bilgiler:

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **20.03.2019 saat 06:59'a** kadar <https://goo.gl/forms/jZTMw9HrifAAOnsf1> adresi üzerinden **HW1_OgrenciNumarasi.rar** dosyasını yükleyiniz.

- Ödev kontrolleriniz **20.03.2019 Çarşamba günü** dersin laboratuvar saatinde yapılacaktır. Detaylı bilgi için Arş. Grv. Ahmet ELBİR'in avesis.yildiz.edu.tr adresindeki sayfasını takip ediniz.

Teslim Edilecekler:

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız. **Yöntem** bölümünde problemi ve çözüm için önerdiğiniz yöntemi kısaca anlatıp rekürsif fonksiyonunuzun durma ve rekürsif olarak çağırılma koşullarını belirtiniz.
2. **Uygulama** bölümünde son taşa ulaşılabilen ve ulaşılamayan 4 ayrı örnek üzerinde algoritmanızın çalışmasını adım adım gösteriniz.
3. Algoritmanızın **C** dilinde programını hazırlayarak dokümana da ekleyiniz.
4. **Teslim Edilecekler**
 - a. HW#_OgrenciNumarasi.rar (Örn: HW1_15011001.rar)
 - i. OgrenciNumarasi.pdf (Örn: 15011001.pdf)
 - ii. OgrenciNumarasi.c (Örn: 15011001.c)

Değerlendirme

Algoritma Tasarımı ve Programın Çalışması: (%80)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Program hatasız çalışmalıdır.
4. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%20)

1. Raporun kapak sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
3. Değişken isimleri anlamlı olmalıdır.
4. Her fonksiyonun yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
5. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
6. Gereksiz kod tekrarı olmamalıdır.
7. Kaynak kodun formatı düzgün, okunabilir ve takip edilebilir olmalıdır.