

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK II REPORT

STUDENT NAME
LÜTFULLAH TÜRKER

STUDENT NUMBER
141044050

Course Assistant:
Nur Banu Albayrak

1. Problem Solutions Approach

2. Test Cases

3. Running and Results

1. Problem Solutions Approach

Q1)

1.

Dıştaki for döngüsü $i = 0$ 'dan $(n-1)$ 'e kadar gittiği için $(n-1)-0$ yani $n-1$ kere çalışır.

İçteki yani ikinci for döngüsü de $j=i+1$ den n ' e kadar gittiği için $n-(i+1)$ kere çalışır.

İçteki döngü dıştaki döngü sayısı kadar çalışacağı için toplam çalışma sayısı :

$(n-1) * (n-i-1) = n^2 - i*n - n - n + i + 1$ ' dir. Her seferinde 3 işlem gerçekleştireceği için bu sonucu 3 ile çarparsak toplam yapılan işlem sayısını buluruz. Sonuç olarak bu durumda gösterim $\theta(n^2)$ olmalıdır.

2.

Bu soru için farklı ihtimaller olduğundan best case ve worst case den bahsedebiliriz.

Best case için :

Boş bir string gelirse direk ilk if kısmına girer ve 0 return eder. Yani 1 kere çalışır. $\theta(1)$

Worst case için :

`return 1 + length(str.substring(1));` satırı için 2 işlem yapılır.

$$T(n) = 2 + T(n-1)$$

$$T(n-1) = 2 + T(n-2)$$

$$T(n-m) = 2 + T(n-m-1)$$

Buradan $T(n) = 2*m$ bulunur ve $m = n$ için $T(n) = 2*n$ olur.

n uzunluğundaki bir String için fonksiyon n kere recursive yapar. Yani n kere çağrılır. $\theta(n)$

Sonuç olarak $O(n)$ dir.

Q2)

1.

Küçükten büyüğe doğru sıralama algoritmasıdır. Her seferinde en küçük elemanı bularak ilk döngüdeki elemana atayarak sonunda ilk başta en küçük olacak şekilde sıralamış olur.

2.

Best Case :

$\theta(n)$ ' dir. En iyi durum için bir döngüde 1 kere diğerinde n kere çalışır.

Worst Case :

$\theta(n^2)$ ' dir. İki döngünün tekrar sayıları toplandığında ikinci dereceden bir denklem elde ederiz. $\rightarrow (n-1)*n/2$

Denklemin en yüksek katsayılı terimi n^2 olduğundan $\theta(n^2)$ buluruz.

Q3)

Bir algoritmanın çalışma süresini üstten ve alttan sınırlamak istediğimizde θ gösterimini kullanırız. Sadece üstten sınırlamak istediğimizde big-O notasyonunu kullanırız ve bu notasyonun anlamı çalışma süresi en fazla bu kadar olabilir demektir. Tam tersi şekilde Omega(Ω) notasyonunda da algoritma en az bu kadar sürede çalışabilir deriz. Yani O notasyonu maximum süreyi Ω notasyonu da minimum süreyi verir.

Bu bilgilere dayanarak bir algoritmanın çalışma süresi $\theta(n)$ ise bu sadece çalışma süresi maximum $O(n)$ sürede ve minimum da $\Omega(n)$ sürede olduğunda mümkündür. Çünkü θ notasyonunda notasyonda belirtilenden başka sürede çalışma ihtimali yoktur.

Matematiksel olarak da :

$\Omega = c_1 * f(n) \leq T(n)$ $O = T(n) \leq c_2 * f(n)$ ikisini birleştirdiğimizde

$\Theta = c_1 * f(n) \leq T(n) \leq c_2 * f(n)$ olur.

Q4)

1.

```
public int InsertionSort(int max) {  
    if (max <= 1)  
        return max;  
    max = InsertionSort(max - 1);  
  
    // maximum elemanı diğer elemanlarla kıyaslıyoruz.  
    for (int i = max - 1; i >= 0 && (arr[i] > arr[max]); --i)  
        arr[i+1] = arr[i];  
    arr[i+1] = arr[max];  
    return max + 1;  
}
```

2.

Worst case'de n elemanlı bir array in sıralama süresi , $(n-1)$ elemanlı bir arrayin sıralama süresi + $(n-1)$ karşılaştırma daha fazladır.Yani $T(n) = T(n-1) + n-1$

Bu şekilde bir recurrence yazarsak :

$$T(n) = T(n-1) + n-1$$

$$T(n-1) = T(n-2) + n-2$$

$$T(n-k) = T(n-k-1) + n-k-1 \quad \dots\dots\dots$$

$$T(2) = T(1) + 1$$

3.

Bu şekilde olan denklemleri iki taraflı olarak toplarsak şunu elde ederiz.

$$T(n) + T(n-1) + T(n-2) + \dots T(4) + T(3) + T(2) = T(n-1) + T(n-2) + \dots T(3) + T(2) + T(1) + (n-1) + (n-2) + \dots + 4 + 3 + 2 + 1$$

Bu denklemde de gerekli sadeleştirmeleri yaptığımızda :

$$T(n) + \cancel{T(n-1)} + \cancel{T(n-2)} + \dots \cancel{T(4)} + \cancel{T(3)} + \cancel{T(2)} = \cancel{T(n-1)} + \cancel{T(n-2)} + \dots \cancel{T(3)} + \cancel{T(2)} + T(1) + (n-1) + (n-2) + \dots + 4 + 3 + 2 + 1$$

$$\Rightarrow T(n) = T(1) + n*(n-1)/2$$

$$\Rightarrow T(n) = 1 + n*(n - 1)/2$$

$$\text{Sonuç olarak işlemlerimizin sonucu} \quad \Rightarrow \quad T(n) = O(1 + n*(n - 1)/2) \quad \Rightarrow \quad O(n^2)$$

Q5)

1.

$$f(n) = n^{0.1} , g(n) = (\log(n))^{10}$$

$f(n) = O(g(n))$ olması için :

$f(n) \leq c*g(n)$ olmalı:

$$n^{0.1} \leq c*(\log(n))^{10} \quad \text{iki tarafın log. sını alırsak : } 0.1*\log(n) \leq c*\log(\log(n))$$

logaritma fonksiyonu her zaman lineer bir fonksiyondan daha yavaş büyür.

Bu yüzden $f(n) = O(g(n))$ diyemeyiz.

$f(n) = \Omega(g(n))$ olması için $f(n) \geq c*g(n)$ olmalı :

$$n^{0.1} \geq c*(\log(n))^{10} \quad \text{bu durumu sağlar.Çünkü lineer kısım logaritmadan daha hızlı büyür.}$$

Yani $f(n) = \Omega(g(n)) \Rightarrow n^{0.1} = \Omega((\log(n))^{10})$ dir.

2.

$$f(n) = n! , g(n) = 2^n$$

$f(n) = O(g(n))$ olması için $f(n) \leq c \cdot g(n)$ olmalı:

$n! \leq c \cdot 2^n$ $c = 1$ ve $n_0 = 4$ için 4 den büyük tüm n değerleri bu durumu sağlar. Dolayısıyla faktöriyelli kısım her zaman daha büyük ve daha hızlı büyüyeceğinden $f(n) = O(g(n))$ diyemeyiz.

$f(n) = \Omega(g(n))$ olması için $f(n) \geq c \cdot g(n)$ olmalı :

$n! \geq c \cdot 2^n$ bunu yukarıda görüldüğü gibi sağlar. **Yani $f(n) = \Omega(g(n)) \Rightarrow n! = \Omega(2^n)$ dir.**

3.

$$f(n) = (\log(n))^{\log(n)} , g(n) = 2^{(\log_2 n)^2}$$

$f(n) = O(g(n))$ olması için $f(n) \leq c \cdot g(n)$ olmalı:

$(\log(n))^{\log(n)} \leq c \cdot 2^{(\log_2 n)^2}$ durumunda exponential taraf logaritma dan kat kat daha hızlı büyüdüğü için örneğin

$C = 1$ ve $n = 10$ için sol taraf 1 iken sağ taraf 2^9 dan daha büyük bir sayı çıkıyor. Yani sağ taraf daha büyük oluyor ve bu eşitsizliği sağlıyor. Bu durumda **$f(n) = O(g(n)) \Rightarrow (\log(n))^{\log(n)} = O(2^{(\log_2 n)^2})$ dir diyebiliriz.**

Q6)

Farklı Array yapılarını kullanarak performans hesaplama problemine yaklaşımım programın çalışma süresini hesaplayarak performansı tespit etmek oldu. Programda kullandığım 3 Arrayi tek tek ArrayList ve LinkedList yaparak çalışma sürelerini hesapladım. Nasıl hesapladım ?

Programın başına o anki milisaniyeyi veren fonksiyonu çağırarak sonucunu start isimli değişkene atadım. Ve programın sonunda da finish değişkeni tanımlayıp finish 'e o anki milisaniyeden start ı çıkarıp atadım. Sonuç olarak finish değişkeninde program başlamasından bitmesine kaç milisaniye çalıştıysa o kadar yazdı ve bunu ekrana bastım.

Programın bu halinde süre önemli olduğu için maindeki kullanıcıdan input alma işlemlerini iptal edip yerine hazır inputlar yazıp test ettim. Asıl main i çok bozmadan bunu yaptığım için biraz anlaşılması zor olabilir.

2. Test Cases

Senaryo	Test	Beklenen Sonuç	Gerçek Sonuç
Program açıldığında sistem süresi milisaniye olarak alınır ve değişkene atılır. Program biterken de sistem süresi tekrar alınır ve başka değişkene atılır.	Program çalışırken hata meydana gelirse ne olur ?	Hata alınana kadar geçen süreyi elde etmek.	Program bittiğinde geçen süre hesaplandığı için ve program sona erişemediği için çalışma süresi hesaplanamaz.
Programın çalışma süresi sistemin zamanını milisaniye cinsinden alan fonksiyonlar ile hesaplanmaktadır.	Program çalışırken kullanıcıdan input almak için beklenirse ne olur ?	Kullanıcının girişi için beklenen süre hesaba katılmadan çalışma süresinin hesaplanması beklenir.	Beklenen sonucun mümkün olması için kullanıcıdan input alınan her yerin öncesinde ve sonrasında tekrar süreyi alıp birbirinden çıkarılarak o sürenin sayılmaması gibi uzun işlemler gerektirir. Asıl durumda kullanıcının beklendiği süre de dahil edildiği için çalışma süresi doğru olarak hesaplanamaz .Bu yüzden bunu test ederken kullanıcıdan input almıyoruz.

3. Running and Results

Array ,ArrayList ve LinkedList kullanarak programımı test ettiğimde aldığım sonuçlar :

```
osboxes@osboxes: ~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src
Adding a new user operation is successfully completed.
New user's ID : 10002 New user's Name : Deneme

The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
Sorry, You have already taken this book.

The Program Running Time in Millisecond is 27
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ java Main
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
1
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
20001
Hello Test Staff
For show all registered users in the system : Press '1'
For add a new user : Press '2'
For add a book : Press '3'
For remove a book : Press '4'
For exit : Press '-1'
2
Please enter a Username for new user .
Deneme
Adding a new user operation is successfully completed.
New user's ID : 10003 New user's Name : Deneme

The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
Sorry, You have already taken this book.

The Program Running Time in Millisecond is 37
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$
```

Doğal olarak her çalıştırmada farklı sonuçlar aldığımdan dolayı kesin bir yorum yapmak mümkün değil.Daha net bir sonuç için biraz daha fazla dolu bir Database(.csv) dosyasıyla çalışmak gerekirdi.Fakat yine de aldığım sonuçlara göre aralarında en yavaşının LinkedList olduğunu gördüm.Bunun sebebini Search operasyonunu sıklıkla yaptığım için olduğunu düşünüyorum.Çünkü LinkedList de arama operasyonu $O(n)$ sürerken ArrayList de $O(1)$ sürüyor.Ve programımda giriş yaparken kullanıcı ID sorgulamasında Kitap adı sorgulamasında gibi çok fazla yerde arayıp kontrol etme işlemi yaptığım için bu fark meydana geliyor diye düşünüyorum.

```
osboxes@osboxes: ~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src
Adding a new user operation is successfully completed.
New user's ID : 10004 New user's Name : Denene

The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
Sorry, You have already taken this book.

The Program Running Time in Millisecond is 28
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ java Main
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
1
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
20001
Hello Test Staff
For show all registered users in the system : Press '1'
For add a new user : Press '2'
For add a book : Press '3'
For remove a book : Press '4'
For exit : Press '-1'
2
Please enter a Username for new user .
Denene

Adding a new user operation is successfully completed.
New user's ID : 10005 New user's Name : Denene

The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
Sorry, You have already taken this book.

The Program Running Time in Millisecond is 27
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$
```

ArrayList kullandığımda ise bu sonucu elde ettim. LinkedList den daha iyi bir sonuç var. Yukarıda belirttiğim sebepten dolayı olsa gerek. Fakat çok fazla fark yok çünkü listeye eleman ekleme çıkarma işlemlerini de az da olsa yapıyoruz. Ekleme çıkarma işlemlerinde de LinkedList ($O(1)$) ArrayListten ($O(n)$) daha hızlı olduğu için bu şekilde aradaki fark biraz kapanıyor diye düşünüyorum.

Kendim Array tanımlayıp boyutunu ayarladığımda ise ArrayList'den çok da farklı bir sonuç almadım. ArrayList de aynı şekilde aynı işlemleri yaptığından dolayı bu işlemleri kendim yapınca da sonuç çok değişmedi.


```
osboxes@osboxes: ~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ javac *.java
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ java Main
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
1
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
20001
Hello Test Staff
For show all registered users in the system : Press '1'
For add a new user : Press '2'
For add a book : Press '3'
For remove a book : Press '4'
For exit : Press '-1'
2
Please enter a Username for new user .
Deneme
Adding a new user operation is successfully completed.
New user's ID : 10001 New user's Name : Deneme
The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
The book borrowing process has been successfully completed.
The Program Running Time in Millisecond is 7
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$
```

```
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ javac *.java
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$ java Main
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
1
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
20001
Hello Test Staff
For show all registered users in the system : Press '1'
For add a new user : Press '2'
For add a book : Press '3'
For remove a book : Press '4'
For exit : Press '-1'
2
Please enter a Username for new user .
Deneme
Adding a new user operation is successfully completed.
New user's ID : 10001 New user's Name : Deneme
The Book has been added in the library.
Please log in.
For Users : Press '0'
For Staffs : Press '1'
For Exit : Press '-1'
0
Please Enter Your ID .(FOR TESTING : STAFF ID = '20001')
10001
For learn books you borrowed : Press '1'
For borrow a book : Press '2'
For return a book : Press '3'
For exit : Press '-1'
2
Please enter a book name you want to borrow.
Seftiller
The book borrowing process has been successfully completed.
The Program Running Time in Millisecond is 8
osboxes@osboxes:~/Desktop/Data Structures and Algorithms HW/Data Structures HW1/src$
```

Bu sonuçları programı ilk çalıştırmada aldım.Fakat birinciden sonrakilerde yukardaki sonuçları aldığım için bu ilk çalışma örneklerini dikkate almadım.İlk seferde neden böyle hızlı bir sonuç verdiğini anlayamadım.