

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 02 REPORT

SEYİT AHMET KARACA
141044084

Course Assistant:
Nur Banu ALBAYRAK

Q1

Q1: 1: $\text{for}(\text{int } i=0; i < n-1; i++) \{$
 $\text{for}(\text{int } j=i+1; j < n; j++) \{$
 3 simple statements
 }
 }

for loop $O(n^2 - n)$ don't
 $O(n^2)$ also

2: $C_B(n) = \Theta(1)$ $\text{str} = \text{null}$ veya $\text{str} = ""$

$$C_w(n) = 1 + C(n-1)$$

$$C_w(n-1) = 1 + C(n-2)$$

$$C_w(n-2) = 1 + C(n-3)$$

⋮ ⋮

$$C_w(n-k+1) = 1 + C(n-k)$$

$$n=k$$

$$+ C_w(n-k) = 1 + C(n-k-1)$$

$$C_w(n) = k + C(n-k-1)$$

$$\underline{\underline{C_w(n) = O(n)}}$$

Q2-Q3

Q2: 1: Sayıları küsülden büyüğe doğru sıralayan bir fonksiyondur.

2: Bu fonksiyonun olmazsa olmaz operasyonu "if" operasyonudur. Yani onu saymalıyız.

içteki döngüde $n-1$ defa döner. $O(n-1)$

exchange operasyonu sabit $O(1)$

dıştaki döngü n defa girer. $O(n)$

nested loop olduğu için $O(n^2-n)$ 'den

$O(n^2)$ olur.

Best case: Eleman sayısı 1 olması durumu $O(1)$

Q3: $C_2 \cdot f(n) \leq g(n) \leq C_1 \cdot f(n) \quad n \geq n_0$

$$g(n) = n + 5 \cdot n^{\frac{1}{2}}$$

$$f(n) = n$$

$$L(g(n)): \frac{C_2 \cdot n}{n} \leq \frac{n + 5n^{\frac{1}{2}}}{n}$$

$$C_2 \leq 1 + 5 \frac{n^{\frac{1}{2}}}{n}$$

$$n=1; \quad C_2 \leq 1 + 5 = 6$$

$$n=2; \quad C_2 \leq 1 + \frac{5\sqrt{2}}{2} = 3,53$$

$$\underline{C_2 = 1}$$

$$C_2 = 1 \quad C_1 = 6 \quad n \geq 1$$

$$O(g(n)): \frac{n + 5n^{\frac{1}{2}}}{n} \leq \frac{C_1 \cdot n}{n}$$

$$1 + \frac{5\sqrt{n}}{n} \leq C_1$$

$$n=1; \quad 6 \leq C_1$$

$$\underline{C_1 = 6}$$

Q4

Q4: $i = \text{index}$ size eleman sayısı

Kada her seferinde diz eleman sayısının bir eksiği kadar karşılaştırma yapıyor.
ve her seferinde fonksiyon eleman sayısının bir eksiği olarak yeniden çağırılıyor.

2: Best case: size ve index eşit olması $\Theta(1)$

Worst case: Dizinin büyükten küçüğe doğru sıralı olması n kere fonksiyon kendini yineliyor. $n-1$ kere karşılaştırma.
 $O(n^2 - n)$ oluyor. $O(n^2)$

3: n eleman sayısı $n = k$

$$T(n) = \Theta(1) \quad n = 0$$

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

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

$$T(n-2) = T(n-3) + n-2$$

$$\vdots$$

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

$$T(n) = n + (n-1) + (n-2) + (n-3) + \dots + 0$$

$$= \frac{n \cdot (n+1)}{2} = \frac{n^2 + n}{2}$$

$$T(n) = O(n^2)$$

Q5

Q5: $f(n) = n^{0.1}$ $g(n) = (\log n)^{10}$
 $n \geq 1$ ve g fonksiyonunun kuvveti f fonksiyonunun
10 katı olduğu için
 $f(n) = O(g(n))$ olur.

$\rightarrow f(n) = n!$ $g(n) = 2^n$
 $n \geq 1$ ve n faktöriyel 2^n 'ye göre çok daha
hızlı artacağı için
 $f(n) = O(g(n))$

$\rightarrow f(n) = (\log n)^{\log n}$ $g(n) = 2^{(\log n)^2}$
$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{(\log n)^{\log n}}{2^{(\log n)^2}} = 0$$

yani $g(n)$ fonksiyonu daha hızlı artar.
 $f(n) = O(g(n))$

Q4'te çalıştığım kod:

```
void InsertionSort(int i, int array[], int size)
{
    if (i < size)
    {
        int j;
        int temp = array[i];

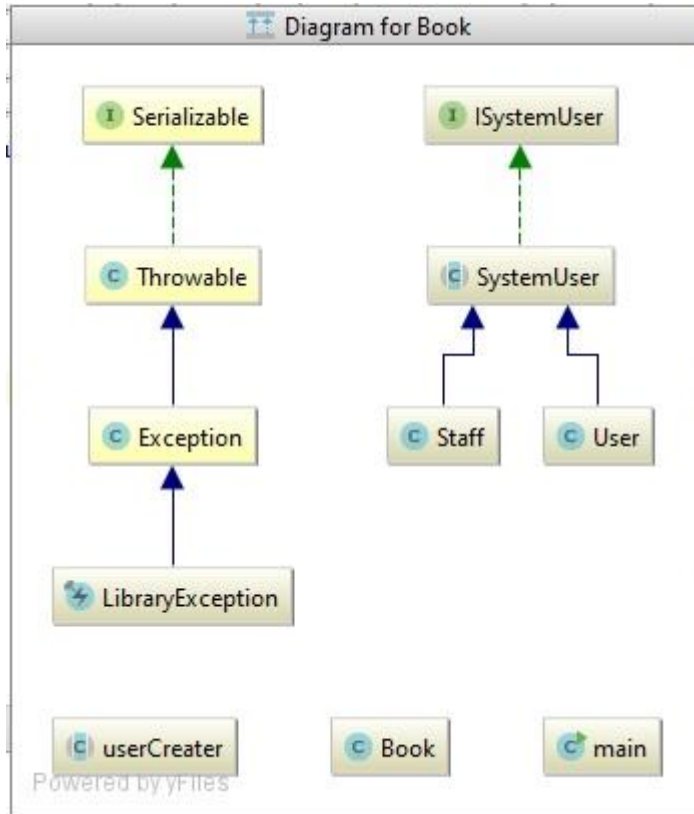
        for (j = i; j > 0 && array[j-1] > temp; j--)
            array[j] = array[j-1];
        array[j] = temp;

        InsertionSort(++i, array, size);
    }
}
```

Linkte 2.postki kullanıcının kodunu kullandım.

<https://cboard.cprogramming.com/c-programming/37188-recursive-insertion-sort.html>

1. Class Diagrams



2. Problem Solutions Approach

Problem ödev 1'deki yazdığım programın array, arralist ve linkedlist ile yeniden implement edilmesi , implement edildikten sonra performansları arasındaki farkların bulunmasıydı. İlk ödevimi arraylist ile yaptığım için önce ilk ödevimi array ile çalışacak şekilde yeniden düzenledim. Sonrasında linkedlist ile çalışabilmesi için tekrar düzenledim ve aralarındaki performanslarını çalışma hızlarına bakarak test ettim. Her yapıda üçer kez methodları çalıştırıp zaman olarak aldığım çalışma sürelerinin ortalamaları ile karşılaştırdım.

3. Test Cases

3.1.Adding book

```
public static long timeAddBook() throws IOException, LibraryException {
    long startTime,stopTime;
    SystemUser su = userCreator.createUser( _ID: "1", _password: "1");
    Staff staff = (Staff)su;

    startTime = System.currentTimeMillis();
    for(int i = 0; i < 1000 ; i++)
        staff.addBook(Integer.toString(i), _name: "a", _autor: "b");
    stopTime = System.currentTimeMillis();

    return stopTime-startTime;
}
```

3.2.Removing book

```
public static long timeRemoveBook() throws IOException, LibraryException {
    long startTime,stopTime;
    SystemUser su = userCreator.createUser( _ID: "1", _password: "1");
    Staff staff = (Staff)su;

    startTime = System.currentTimeMillis();
    for(int i = 0; i < 1000 ; i++)
        staff.removeBook(Integer.toString(i));
    stopTime = System.currentTimeMillis();

    return stopTime-startTime;
}
```

3.3.Adding user

```
public static long timeAddUser() throws IOException, LibraryException {
    long startTime,stopTime;
    SystemUser su = userCreator.createUser( _ID: "1", _password: "1");
    Staff staff = (Staff)su;

    startTime = System.currentTimeMillis();
    for(int i = 3; i < 1003 ; i++)
        staff.addUser(Integer.toString(i), _name: "a", _surname: "b", _type: false, _passowrd: "0");
    stopTime = System.currentTimeMillis();

    return stopTime-startTime;
}
```

3.4.Removing user

```
public static long timeRemoveUser() throws IOException, LibraryException {
    long startTime,stopTime;
    SystemUser su = userCreator.createUser( _ID: "1", _password: "1");
    Staff staff = (Staff)su;

    startTime = System.currentTimeMillis();
    for(int i = 3; i < 1003 ; i++)
        staff.removeUser(Integer.toString(i));
    stopTime = System.currentTimeMillis();

    return stopTime-startTime;
}
```

Methodlarda herhangi bir errorhandling oluşturulmadan kullanıldı
SystemUser.csv de normal ve personel olarak iki tane sistem kullanıcısı kayıtlı olması gerekmektedir..

7. Running and Results

Hw1'in array implement edilmiş hali.

```
Array
Kitap ekleme suresi :2.403 sn
Kitap silme suresi :2.707 sn
Kullanici ekleme suresi :2.668 sn
Kullanici silme suresi :2.658 sn
kitap odunc alma suresi :3.338 sn
kitap odunc verme suresi :3.278 sn
```

Hw1'in arraylist implement edilmiş hali.

```
ArrayList
Kitap ekleme suresi :3.281 sn
Kitap silme suresi :2.725 sn
Kullanici ekleme suresi :3.344 sn
Kullanici silme suresi :2.829 sn
kitap odunc alma suresi :3.6 sn
kitap odunc verme suresi :3.48 sn
```

Hw1'in linkedlist implement edilmiş hali.

```
Linkedlist
Kitap ekleme suresi :2.484 sn
Kitap silme suresi :3.171 sn
Kullanici ekleme suresi :3.963 sn
Kullanici silme suresi :3.104 sn
kitap odunc alma suresi :6.272 sn
kitap odunc verme suresi :6.326 sn
```

Her yapıda fonksiyonları biner kullanıcı ve kitap ekleme,silme ve kitap odunc alma geri iade etme işlemleri yapıldı.Arraylist ve linkedlistin aynı methodları kullanıldı.Member methodlarda kullandığım diğer komutlar ikisinde de aynı olmasına rağmen linkedlistin remove methodu linkedliste göre daha yavaş çalışıyor.Array de ise sadece yer ayırmak için ek fonksiyon yazdım onun dışında indexleri kullanarak işlemler yaptım.Genel olarak bakıldığında bu ödevimin array ile implementasyonunda array diğer iki yapıya göre daha hızlı çalışıyor.

Github: https://github.com/SeyitAhmetKARACA/141044084_HW02/