Onur Sezer
121044074

CSE-321 HW01

1-) Prove the statements below whether it is true or not.

a) It is true for all $f(n)$ and $g(n)$ functions

$f(n) \in O(g(n))$ or $g(n) \in O(f(n))$

Solution:

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$, sıfır çıktığında kuraldan

$f(n) \in O(g(n))$ 'olur ancak $g(n) \in O(f(n))$ 'nin de doğru olması için $\lim_{n \to \infty} \frac{g(n)}{f(n)} = 0$ çıkması gerekiyor ikisi aynı anda sağlanamayacağı için bu şık yanlıştır.

b) Let $f(n)$, $g(n)$ and $h(n)$ be functions. If $f(n) \in O(g(n))$ then the following is true:

$\frac{f(n)}{h(n)} \in O\left(\frac{g(n)}{h(n)}\right)$

Solution:

$f(n) \in O(g(n))$ ise $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$ 'dır.

$\frac{f(n)}{h(n)} \in O\left(\frac{g(n)}{h(n)}\right) \longrightarrow \lim_{n \to \infty} \frac{f(n)}{\cancel{h(n)}} \cdot \frac{\cancel{h(n)}}{g(n)} = \lim_{n \to \infty} \frac{f(n)}{g(n)}$ 'de

yukarda 0 bulduğumuz için $\frac{f(n)}{h(n)} \in O\left(\frac{g(n)}{h(n)}\right)$ doğrudur.

c) Let $f(n)$ and $g(n)$ be functions and k be an integer. If $f(n) \in O(g(n))$ then the following is true:
$f(n)^k \in O(g(n)^k)$

Solution:

$f(n) \in O(g(n))$ ise $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ 'dır

$f(n)^k \in O(g(n)^k) \rightarrow \lim_{n\to\infty} \frac{f(n)^k}{g(n)^k} = \lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right)^k = 0$ 'dır

Bu yüzden $f(n)^k \in O(g(n)^k)$ doğrudur

2.) Prove or disprove the following statements

a. $n^3 \in O(2^n)$

$\lim_{n\to\infty} \frac{n^3}{2^n} = 0$, $n^3 \in O(2^n)$ doğru

b. $2^n \in o(3^n)$

$\lim_{n\to\infty} \frac{2^n}{3^n} = 0$, $2^n \in o(3^n)$ doğru

c. $n! \in O(100^n)$

$\lim_{n\to\infty} \frac{n!}{100^n} = \infty$, $n! \in O(100^n)$ yanlış

3.) Write the pseudocode of linear search with repeated elements. Analyze its best case, worst case, and average case complexities.

Solution:

Pseudocode:

n elemanlı liste

```
function linearSearch ( L[1:n], x )
    for i=1 to n do
        if ( L[i] == x ) then
            return i
        end if
    end for
```

aranan eleman

Best case: Aranan eleman listenin ilk sırasında bulunursa, tek karşılaştırma yapmış olur. Bu yüzden complexitie'si $B(n)=1 \in O(1)$ bulunur.

Worst case: Aranan eleman listenin son sırasında bulunursa n defa karşılaştırma yapmış olur. Bu yüzden complexitie'si: $w(n)=n \in O(n)$ bulunur.

Average case:

$$\text{average} = (1+2+3+4 \dots +n)/n$$

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

$$= \frac{n+1}{2} \in O(n)$$

4) Enigma algorithm is given below

```
Algorith Enigma (A[0...n-1, 0, ...n-1])
// input the matrix A of real numbers
for i=0 to n-2 do
    for j=i+1 to n-1 do
        if A[i,j] != A[j,i]
            return false
return true
```

a) what is the problem size (input size)?

input size : n

b) what is the main operation of the algorithm?

main operation : if A[i,j] != A[j,i]

simetrik olup olunmadığına bakıldığı if kodu main operation' dur.

c) what is the calculated by this algorithm?

Matrix'in simetrik olup olunmadığına bakılır. Simetrik ise 'true' değilse 'false' return eder.

d)

i- worst case scenario :

worst case durumunda $n^2/2$ defa karşılaştırma olcak, bu yüzden worst case $O(n^2)$ olur.

ii- best case scenario :

best case durumunda ise ilk karşılaştırmada simetrik olmadığının bulunduğu durumdur. O yüzden best case $O(1)$ bulunur.

iii - Average case scenario

average case : $\frac{n^2}{2}$ $O(n^2)$

5.) You are given the following function

```
Function myFun2 (A[1...n])
// Input      : an array of integers
// Output     : an integer
   if (n <= 1)
       return 4
   else
       return myFun2 (A[1...n/3] + myFun2 (A[2n/3 ...n])
```

a) What is problem size ?

problem size : n

b) ?

c) ?

d) ?

6.) Let A be a set, n be an even number, and A consist of n positive integers. Design an efficent algorithm to separate set A into set A_1 and set A_2, each of which includes $n/2$ items, in order to maximize the difference between the addition of elements in set A_1 and the addition of elements in set A_2. Analyze and show your algorithm's complexity using proper asyptotic notations.

Solution :

```
for i:=0 to n do
    if ( i < n/2 ) then
        A1[i] = A[i]
    else
        A2[i-n/2] = A[i]
    endif
endfor
int sumA1 =0 , sumA2 =0, diff
for i:=0 to n/2 do
    sumA1 += A1[i]
endfor
for i:=0 to n/2 do
    sumA2 += A2[i]
endfor
diff = sumA1 - sumA2
```

n (first loop), n/2 (second loop), n/2 (third loop)

complexity $\rightarrow n + \frac{n}{2} + \frac{n}{2} = 2n \rightarrow \Theta(n)$ gibar

7-) Design a recursive algorithm to find out how many 0s are included in an array which consists of only 0s and 1s. Analyze and show your algorithm's complexity using proper asymptotic notations.

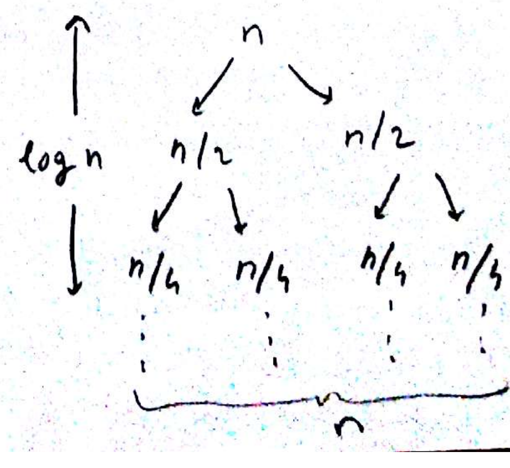
PS: Algorithm is expected to run by dividing an array into two equal parts.

<u>Solution:</u>

```
Function findZero ( A[1...n], size )
// Input   : an array of integers ande size of array
// output  : an integer

    if ( size == 0 )
        return 0
    endif
    else if ( *A == 0 )
        return 1
    end if
    int mid = size / 2
    int rsize = size - mid
    int lsum = findZero( A, mid )
    int rsum = findZero ( A+mid, rsize )
    return lsum + rsum
```

complexity → $\Theta(n \log n)$



8) Solve the recursive equations below. Calculate exact values and explain using theta notation

a) $T(n) = -4 * T(n-1) - 4 * T(n-2)$, $T(0)=0$, $T(1)=1$

$r^2 + 4r + 4 = 0$

$a=1$, $b=4$, $c=4$

$\Delta = b^2 - 4ac$

$\Delta = 16 - 4.1.4$

$\Delta = 0$

$x_{1,2} = \frac{-b \mp \sqrt{\Delta}}{2a} = \frac{-4 \mp \sqrt{0}}{2.1}$

$x_1 = -2$ $x_2 = -2$

$T(n) = c_1 (-2)^n + c_2 (-2)^n$

$c_1 + c_2 = 0$

$-2c_1 + 2c_2 = 1$

$\boxed{O(1)}$

b) $T(n) = T(n-1) + 6 * T(n-2)$, $T(0)=3$, $T(1)=6$

$r^2 = r + 6$

$r^2 - r - 6 = 0$

$(r-3).(r+2)$

$T(n) = c_1 3^n + c_2 (-2)^n$

$T(0) = c_1 + c_2 = 3$ $/2$

$T(1) = 3c_1 - 2c_2 = 6$

$5c_1 = 12$

$c_1 = 12/5 = 2,4$

$c_2 = 0,6$

$T(n) = 2,4.3^n + 0,6.(-2)^n$

$\boxed{\Theta(3^n)}$

c) $T(n) = -5 * T(n-1) - 6 * T(n-2) + 42 * 4^n$, $T(1)=56$, $T(2)=278$

9) Give exact solutions for $T(n)$ in each of the following recurrences.

a) $T(n) = T(n-1) + (n^2+1)$, $T(0)=3$

$n = n-1 \rightarrow \quad T(n-1) = T(n-2) + ((n-1)^2+1)$

$n = n-2 \rightarrow \quad T(n-2) = T(n-3) + ((n-2)^2+1)$

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

$T(n) = T(n-3) + (n-2)^2 + 1 + (n-1)^2 + 1 + n^2 + 1$

$T(n) = T(n-3) + (n-2)^2 + (n-1)^2 + n^2 + 3$

$T(n) = T(n-3) + 3 + (n-2)^2 + (n-1)^2 + n^2$

$T(n) = T(n-k) + k + \frac{k.(k+1).(2k+1)}{6} \longrightarrow \boxed{\begin{array}{l} n-k=0 \\ n=k \end{array}}$

$= T(0) + n + \frac{n.(n+1).(2n+1)}{6}$

$= \boxed{n + \frac{n.(n+1).(2n+1)}{6} + 3}$

b)

c)

10) ?