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

O notation $\rightarrow$ $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ $c > 0$ $n > 0$

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

$0 \leq f(n) \leq c_1 \cdot g(n)$

$f(n) \leq c_1 \cdot g(n)$

$\frac{f(n)}{c_1} \leq g(n)$ doğru olması için $\frac{g(n)}{f(n)} \geq \frac{1}{c_1}$ olmalıdır. Ancak

eğer $g(n) = 0$ olur ve $c_1 = 1$ olursa denklem

sağlanmaz.

2) $g(n) \in O(f(n))$ $\frac{f(n)}{g(n)}$

$0 \leq g(n) \leq c_1 \cdot f(n)$

$g(n) \leq c_1 \cdot f(n)$

$\frac{g(n)}{f(n)} \leq c_1$ $c_1 = 1$ $g(n) = 5$ $f(n) = 1$ için denklem sağlanmamış

oluyor

It is not true.

b.) Let f(n) and g(n) be functions. If $f(n) \in O(g(n))$ then the following is true: $\frac{f(n)}{g(n)} \in O\left(\frac{f(n)}{g(n)}\right)$

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

$0 \leq f(n) \leq c_1 \cdot g(n)$

$\frac{f(n)}{g(n)} \leq c_1$ ①

$0 \leq \frac{f(n)}{g(n)} \leq c_2 \cdot \frac{f(n)}{g(n)}$ ②

$0 \leq c_1 \leq c_2 \cdot c_1 \rightarrow$ This is true.

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: $f(n)^k \in O(g(n)^k)$

1.yol' $f(n) \in O(g(n))$

$0 \leq f(n) \leq c_1 \cdot g(n)$ $\qquad \frac{f(n)}{g(n)} \leq c_1 \qquad c_1 > 0$

$0 \leq f(n)^k \leq c_2 \cdot g(n)^k$ $\qquad \left(\frac{f(n)}{g(n)}\right)^k \leq c_2 \qquad c_2 > 0$

$(c_1)^k \leq c_2$ it is true

2.yol $\lim\limits_{n \to \infty} \frac{f(n)}{g(n)} \overset{\text{L-hopital}}{=} \frac{f'(n)}{g'(n)} = 0$

$\lim\limits_{n \to \infty} \frac{f(n)^k}{g(n)^k} = \frac{k \cdot f'(n)^{k-1}}{k \cdot g'(n)^{k-1}} \rightarrow \left(\frac{f'(n)}{g'(n)}\right)^{k-1} \Rightarrow 0^{k-1} = 0,\ f(n)^k \in O(g(n)^k)$

Scanned by CamScanner

2. Prove or disprove the following statements.

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

$0 \leq n^3 \leq c_1 \cdot 2^n$ $\quad c_1 > 0 \quad n_0 > 0 \quad n \geq n_0$

$n^3 \leq c_1 \cdot 2^n$ $\quad c_1 = 1 \quad n = 1$ için $1 \leq 2$ doğrudur

$n = k$ için $\quad k^3 \leq c_1 \cdot 2^k$ doğru kabul edilir.

$n = k+1$ için $\quad (k+1)^3 \leq c_1 \cdot 2^{k+1}$

$(k+1)^3 \leq c_1 \cdot 2^k \cdot 2$

$\dfrac{(k+1)^3}{2} \leq c_1 \cdot 2^k$

$\dfrac{(k+1)^3}{2} \leq k^3$

$k^3 + 3k^2 + 3k + 1 \leq 2k^3$

$-k^3 + 3k^2 + 3k + 1 \leq 0$

$k^3 - 3k^2 - 3k - 1 + 6k - 6k \geq 0$

$k^3 - 3k^2 + 3k - 1 \geq 6k$

$(k-1)^3 \geq 6k$

$\dfrac{(k-1)^3}{k} \geq 6$ $\quad k = 4$ için denklem sağlanır.

3.) ...

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

$0 \leq f(n) < c.g(n)$

$0 \leq 2^n \leq 3^n . c_1$

$\frac{2^n}{3^n} \leq c_1 \Rightarrow \left(\frac{2}{3}\right)^n \leq c_1$ $n=1$ $c_1=1$ için doğrudur.

$n=k$ $\left(\frac{2}{3}\right)^k \leq c_1$ için doğru kabul edilir.

$n=k+1$ $\left(\frac{2}{3}\right)^{k+1} \leq c_1$ $\left(\frac{2}{3}\right)^k . \frac{2}{3} \leq c_1$

$\left(\frac{2}{3}\right)^k \leq c_1 . \frac{3}{2}$

$c_1 \leq c_1 . \frac{3}{2}$ ✓ doğruluğu kanıtlanmıştır.

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

$0 \leq f(n) \leq g(n).c_1$

$0 \leq n! \leq 100^n.c_1$ $n=1$ $c_1=1$ için doğrudur.

$n=k$ için $k! \leq 100^k . c_1$ $\frac{k!}{100^k} \leq c_1$

$n=k+1$ için $(k+1)! \leq 100^{k+1} . c_1$

$(k+1).k! \leq 100^k . 100 . c_1$ $\frac{k!}{100^k} \leq \frac{100}{k+1} . c_1$

$c_1 \leq \frac{100}{k+1} . c_1$

$\frac{100}{k+1} \geq 1$ $k \leq 99$

3.) Write the pseudocode of linear search with repeated elements. Analyze it's best case, worst case, average case complexities.

3 1 1 2 1

Eğer 1 aranırsa fonksiyon 2. döndürür. ilk bulduğu indextir.

```
Function LinearSearch (L[1:n], x)
    for i=1 to n do
        if (L[i] = x) then
            return i;
        end if
    end for
    return -1;
end
```

Best case: Aranan eleman listenin başında ise constant zaman sürer

$B(n) = 1 \in \Theta(1)$

Worst case: Her koşulda döngüye girer ve ya sonda bulur ya da hiç bulamaz.

$\sum_{i=1}^{n} 1 = W(n) = n \in \Theta(n)$

Average case: $\sum_{i=1}^{n} i \cdot P_i$

$0 \leq P < 1$ = basarılı olma olasılığı

X = kaç tane bulunduğu.

$$= \sum_{n=1}^{n-1} i + \frac{P}{n} \cdot X + n \cdot \underbrace{\left(\frac{P}{n} + (1-P) \right)}_{\text{n-1. elemanda olması yada olmaması.}}$$

$$= \cancel{\frac{n \cdot (n-1)}{2} \cdot \frac{PX}{n}} + P + n - nP$$

$P = 0$ için $A(n) \in \Theta(n)$

$P = 1$ için $A(n) \in \Theta\left(\frac{X \cdot (n-1)}{2} + 1 \right)$

Scanned by CamScanner

4. Enigma algorithm is given below.

```
Algorithm Enigma(A[0...n-1], 0...n-1])

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 problem size?

$$\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} 1 = 1.(n-1-i-1+1)$$

$$\sum_{i=0}^{n-2} n-i-1$$

$$= \sum_{i=0}^{n-2} n-1 - \sum_{i=0}^{n-2} i$$

$$= \underbrace{(n-1).(n-2-0+1)}_{n-1} - \sum_{i=1}^{n-1} i+1$$

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

$$= (n-1)^2 - \frac{n.(n-1)}{2} + (n-1) \Rightarrow (n-1).(n-1-\frac{n}{2}+1)$$

$$= (n-1).\frac{n}{2} = \frac{n^2-n}{2}$$

b.) What is the main operation of the algorithm?

Ana işlem if bloğudur. Eğer simetrik değilse return eder ve ana işlem biter.

c.) What is calculated by this algorithm?

2 boyutlu dizinin simetrik olup olmadığı hesaplanır.

d.) Show this algorithms complexity using the O, Ω, Θ asymtotic notations for following states below.

i. Worst case: Eğer bütün liste simetrik ise döngü sonuna kadar gider ve karmaşıklığı input size olur.

$T(n) = \frac{n^2-n}{2}$

Ω

Big O $\Rightarrow$ $0 \leq \frac{n^2-n}{2} \leq c.n^2$

c=1 için $\frac{n^2-n}{2} \leq n^2$

n=1 için $\frac{1-1}{2} \leq 1$ ✓ $O(n^2)$

Omega $\Rightarrow$ $c_1.n^2 \leq \frac{n^2-n}{2}$

c=1 için

$n^2 \leq \frac{n^2-n}{2}$

n=1 için sağlamaz $\cancel{\Omega(n^2)}$

$W(n) \in O(n^2)$

ii) Best case: ilkinde simetri bozulursa best case olur.

$$B(n) = \Theta(1) \qquad c_1 \cdot 1 \leq 1 \leq c_2 \cdot 1$$

iii) Average case: $\sum_{i=0}^{n-2} \cdot \sum_{j=i+1}^{n-1} 1 \cdot P_i$

5.) You are given the following function:

```
Function myFun (A[1 ... n])
    if ( n <= 1)
        return 4
    else
        return myFun (A[1 ... n/3]) + myFun (A[2n/3 ... n])
```

a.) 0^{th} iterasyon : n

1^{th} : $n/3$

2^{th} : $n/9$

i^{th} : $n/3^i$

last : 1

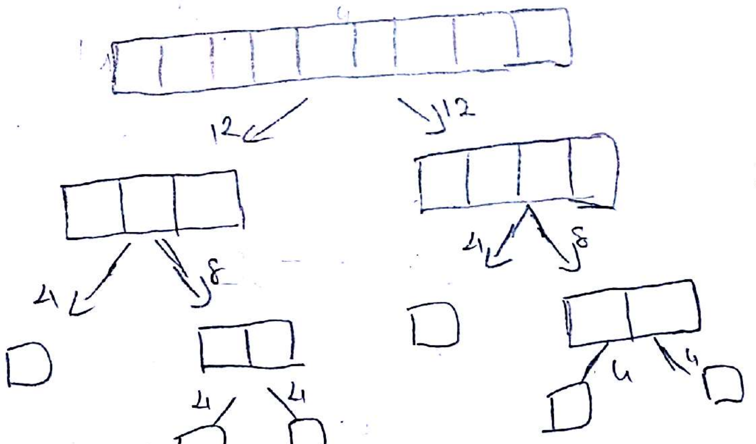
$1 : n/3^i$

$3^i = n$

$\log_3 n = i \Rightarrow$ problem size

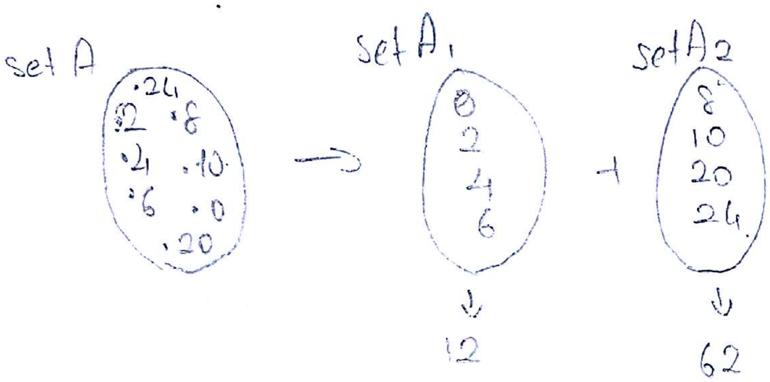
$= \theta(\log_3 n)$

c.)



$= 24$

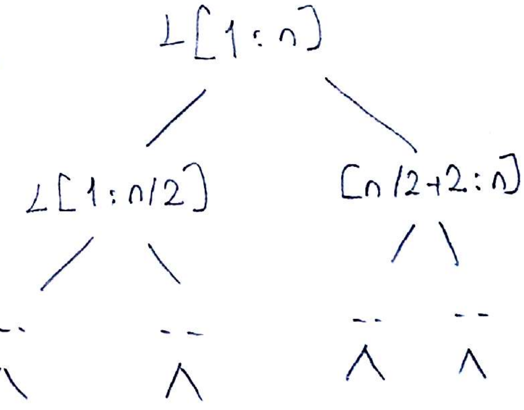
6. Let A be a set, n be an even number, and A consist of n positive integers. Design an efficient algorithm to seperate set A into set A_1 and set A_2, each of which includes $n/2$ items, in order to maximize the difference between the addition of elements in set A_1 and the addition of elements in set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



⇒ aradaki fark → 50

Dolayısıyla ilk olarak verilen listeyi en hızlı şekilde sıralamalıyım. Sonra ortadan ikiye bölerek aralarında maksimum farkı elde ederim. En hızlı sıralama algoritması Quicksort kullanırım.

Quicksort →



+

```
for(i to n/2)
    fill A1
for(n/2+1 to n)
    fill A2
```

$\Theta(n)$

→ Quick Sort için pivot ortadaki eleman seçtim.

Best Case : $(k-1).(n+1)$

$\cancel{n} + n \log n$

$2^{k-1} = n \Rightarrow \log_2(n+1) = k$

$(\log_2(n+1) - 1).(n+1) \Rightarrow \Theta(n.\log_2 n)$

Worst Case : Quicksort icin worstcase = $\Theta(n^2)$'dir.

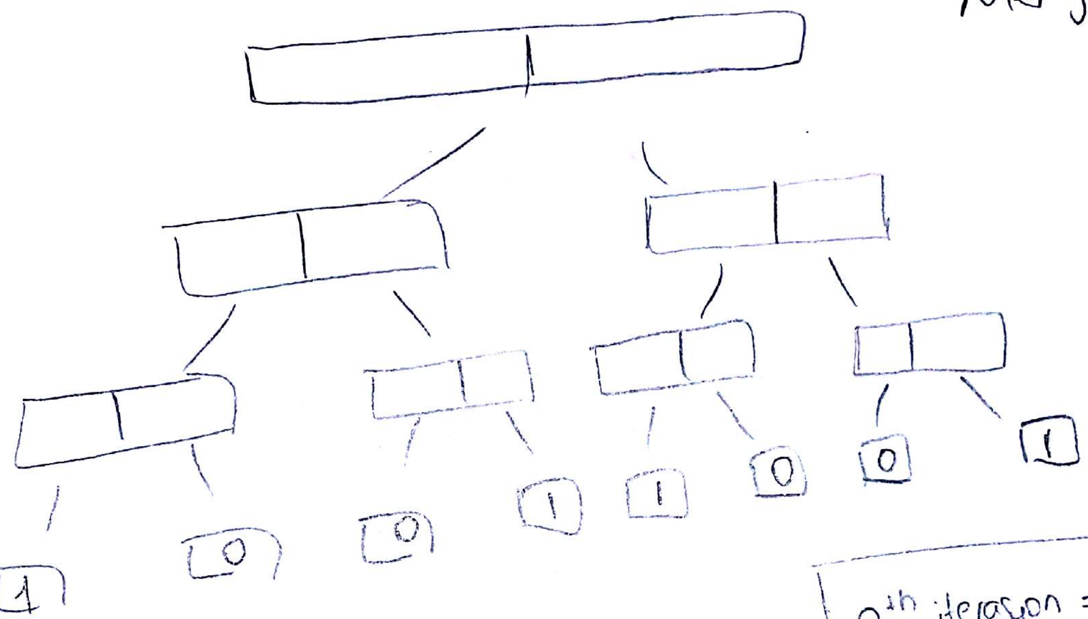
Bu algoritmanın karmasıklığı = $\Theta(n^2)$ olur.

Average Case : Quicksort icin average case = $\Theta(n.\log n)$'dir.

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 asyntotic notations.

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

Merge




```
Function myFun(A[1 ... n]){

    if(n<1)
        return 0;
    if(n==1 && A[1]==0)
        return 1;
    if(n==1 && A[1]==1)
        return 0;

    if(n>1)
        return 1 + myFun(A[1...n/2]) + myFun(A[n/2+1 ... n]);
}
```

0^{th} iterasyon	$= n$
1^{th}	$= n/2$
2	$= n/4$
i^{th}	$= n/2^i$
last	$= 1$

$n/2^i = 1$

$\log_2 n = i$

$\Theta(\log_2 n)$

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

a.) $T(n) = -4 \cdot T(n-1) - 4T(n-2)$, $T(0)=0$, $T(1)=1$

$T(2) = -4T(1) - 4T(0)$

$T(3) = -4T(2) - 4T(1)$

$T(4) = -4T(3) - 4T(2)$

$T(5) = -4T(4) - 4T(3)$

0	0	0 . 0
1	1	1 . 1
2	-4	2 . 2
3	12	4 . 3
4	-32	8 . 4
5	80	

$80 = 16 \cdot 5$

$16 \cdot 5 = 80$

$T(0)=0$
$T(1)=1$
$T(2)=-4$
$T(3)=12$
$T(4)=-32$
$T(5)=80$

formül $= -1^{(n-1)} \cdot 2^{(n-1)} \cdot n$
$= (-2)^{(n-1)} \cdot n$

$$\underline{c_1 \cdot 2^n \leq -2^{(n-1)} \cdot n \leq c_2 \cdot 2^n}$$

$c_1 \cdot 2^n \leq -2^{(n-1)} \cdot n$

$c_1 = 1$ $n=3$ için $8 \leq 12$ sağlar

$n=k$ için $c_1 2^k \leq (-2)^{(k-1)} k$ doğru kabul et.

$n=k+1$ için $c_1 \cdot 2^{k+1} \leq -2^{(k)} \cdot (k+1)$

$c_1 \cdot 2^k \cdot 2 \leq -2^{(k)} \cdot (k+1)$

$(-2)^{(k-1)} \cdot k \cdot 2 \leq (-2)^k \cdot (k+1)$

$-2^{-1} \cdot k \cdot 2 \leq k+1$

$-k \leq k+1$ ✓

$-2^{(n-1)} \cdot n \leq c_2 \cdot 2^n$

$c_2 = 1$ $n=2$ için $-2 \leq 4$ sağlar.

$n=k$ için $-2^{(k-1)} \cdot k \leq c_2 \cdot 2^k$ doğru kabul et

$n=k+1$ için $(-2)^{(k)} \cdot (k+1) \leq c_2 \cdot 2^{k+1}$

$(-2)^k \cdot (k+1) \leq (-2)^{k-1} \cdot k \cdot 2$

$-2^1 \cdot (k+1) \leq 2k$

$-(k+1) \leq k$ ✓

$$\Theta\left((-2)^{(n-1)} \cdot n\right)$$

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

$T(2) = T(1) + 6\,T(0) \Rightarrow 24$

$T(3) = T(2) + 6\,T(1) \Rightarrow 60$

$T(4) = T(3) + 6\,T(2) \Rightarrow 204$

$6 \cdot 2 \cdot 2^{-2}$

$6 \cdot 2 \cdot 2^{-1}$

$6 \cdot 2 \cdot 2$

$6 \cdot 2 \; 5$

$6 \cdot 2 \; 17$

3.1

3.2

3.8

3.20

3.78

180
60
120

$5! + T(4)$

34
78
124
248

84
2
168
24
6
3

27
3
81
3
243
33
210

8.c) $T(n) = 2T(n-1) - T(n-2) + n$ $\quad T(0)=0$

$T(1)=0$

$T(2) = 2T(1) - T(0) + 2$ $\quad T_2 = 2 \quad 2^1$

$T(3) = 2T(2) - T(1) + 3$ $\quad T_3 = 7 \quad 2^3 - 1$

$T(4) = 2T(3) - T(2) + 4$ $\quad T_4 = 16 \quad 2^4$

$T(5) = 2T(4) - T(3) + 5$ $\quad T_5 = 30 \quad 2^5 - 2$

Scanned by CamScanner

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

a.) $T(n) = T(n-1) + (n^2+1) \qquad T(0)=3$

$\cancel{T(n-1)} = T(n-2) + (n-1)^2 + 1$

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

$T(\cancel{1}) = T(0) + 2$

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

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

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

b.) $T(n) = \sum_{i=1}^{n-1} T(i) + n^2 \qquad T(1)=1$

$T(n) =$

10. Solve the recursive equations below asymptotically.

a.) $f(n) = 3f(n/2) + n^2$, $f(1) = 4$

$f(t) = n^2$

$x(n) = a \cdot x\left(\frac{n}{b}\right) + f(n)$

$f(1) = 4$

$c = 4$

$a = 3$. $f(n) = n^2 = \Theta(n^d)$ $d = 2$ //

$b = 2$

$\Theta(n^d)$; if $a < b^d$

$3 < 2^2$ ✓

$= n^d = n^2 \Rightarrow \Theta(n^2)$

b.) $f(n) = 3f(n/2) + n^2 \log n$, $f(1) = 1$

$a = 3$

$b = 2$

$f(n) = n^2 \log n \rightarrow \Theta(n^2)$ $d = 2$ $c = 1$

$a < b^d$

$\Theta(n^2)$ //