① int sequentialSearch(int n, int* A, int x){

```
int i=0;
while(i<n && A[i]!=x)
    i++;
if(i<n)
  return i;
else
  return -1;
}
```

Basic operation

n-1

$C_{best}(n) = 1$ $\quad C_{worst}(n) = \sum_{i=0}^{n-1} 1 \to n-1 \approx n$

$C_{avg}(n)$ → p: $0 \le p \le 1$ i. poz. yerde olması $\frac{p}{n}$ olasılık

$C_{avg}(n) = n.(1-p) + [\frac{p}{n} + \frac{2p}{n} + \frac{3p}{n} \dots \frac{np}{n}]$

⇓ eğer yoksa

varsa → $\sum_{i=1}^{n} \frac{p}{n}.i = \frac{n.(n+1).p}{2n} \to \frac{(n+1)p}{2}$

→ p= 0 için $C_{avg}(n) = n + 0 = n$ p=1 için $C_{avg}(n) = 0 + \frac{n+1}{2} \approx n$

$C_{avg}(n) = n \to O(n)$ $C_{worst}(n) = n \to O(n)$ $C_{best}(n) = 1 \to \Omega(1)$

② $\frac{1}{2}n(n-1) \in \Theta(n^2)$

if $0 \le c_2 * g(n) \le f(n) \le c_1 * g(n)$ and $c_1 \ge c_2, c_1 > 0, c_2 > 0$

then $f(n) \in \Theta(g(n))$

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

$c_2 = \frac{1}{3}$ → $\frac{1}{3}n^2 \le \frac{1}{2}n^2 - \frac{1}{2}n \to n_0 = 25$ → $n_0 = 25$ $c_2 = \frac{1}{3} \to \Omega(n^2)$

$c_1 = 15 \to \frac{1}{2}n^2 - \frac{1}{2}n \le 15n^2 \to n_0 = 25$ → $n_0 = 25$ $c_1 = 15 \to O(n^2)$

$\Theta(n^2)$

$15 > \frac{1}{3}$ $\quad 15 > 0$ $\quad \frac{1}{3} > 0$

(3) a. $\sum_{i=3}^{n+1} i \rightarrow (n+1-3+1)\cdot\frac{(n+1+3)}{2} = (n-1)\frac{(n+4)}{2} = \frac{n^2+3n-4}{2}$

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

$\frac{(n-1)(n)(2n-1)}{6} + \frac{(n-1)(n)}{2} \rightarrow \frac{(2n^2-n)(n-1)+(n-1)(n)\cdot 3}{6}$

$\rightarrow \frac{(n-1)(2n^2+2n)}{6}$

(4) $x(n/2) = x(n/4) + n/2 \qquad x(n/4) = x(n/8) + n/4 \rightarrow x(n) = x(n/8) + \frac{3n}{4}$

$x(n/2) = x(n/8) + \frac{3n}{2} \rightarrow x(n) = x\left(\frac{n}{2^k}\right) + \frac{(2^k-1)n}{2^{k-1}}$

$n = 2^k \qquad k = \log_2 n$

$x(n) = x\left(\frac{2^k}{2^k}\right) + \frac{(2^{\log_2 n}-1)n}{2^{\log_2 n - 1}} \rightarrow \frac{(n-1)n}{\frac{n}{2}} + 1 \rightarrow 2n-2+1 = 2n-1$

(5)

```
dToDecimal(m, number, d)
    digit ← 1                                   -> m+1 control
    sum ← 0                                     -> basic operation
    for i ← 0 to m; do                          -> 4*m operation
        endDigit ← number mod 10
        sum ← sum + endDigit * digit
        digit ← d * digit
        number ← number/10
    return sum
```

$O \rightarrow 5m+1 \leq c_1 * m \rightarrow c_1 = 20 \rightarrow O(n)$

$n_0 = 1$

$\Omega \rightarrow c_2 * m \leq 5m+1 \qquad c_2 = 1 \rightarrow \Omega(n)$

$n_0 = 1$

$c_1 \geq c_2 \qquad c_1 > 0 \qquad c_2 > 0 \qquad n_0 = 1$

$\Theta \leq c_2 * m \leq 5m+1 \leq c_1 * m \rightarrow \Theta(n)$

Ödev 7

DIVIDE AND CONQUER

```
DAC (A[0...n-1], low, high)
    if high ≤ low then
        return A[low]
    mid = (low+high)/2
    sum=0; left_max = min(A);
    for i = mid ; i ≥ low ; i--
        sum = sum + A[i];
        if (sum > left_max)
            left_max = sum;
    right_max = min(A); sum = 0;
    for i=mid+1 ; i ≤ high ; i++
        sum = sum + A[i];
        if (sum > right_max)
            right_max = sum;
    left_right = max( DAC(A[], low, mid), DAC(A[], mid+1, high) )
    return max(left_right, left_max + right_max)
```

$T(n) = 2T(n/2) + N \rightarrow d=1,\ a=2,\ b=2 \qquad 2 = 2^1 \rightarrow n^d \log n \rightarrow n \log n$

Her durumda tüm elemanları dolaşmak zorunda olduğu için

$C_{best} = \Theta(n \log n) \qquad C_{average} = \Theta(n \log n) \qquad C_{worst} = \Theta(n \log n)$

BRUTE FORCE

```
DF(N,A[N])
    max = A[0]
    for  i=0; i<N; i=i+1
        tmp=0
        for j=i; j<N; j=j+1
            tmp=tmp+A[j]
            if(tmp > max)
                max = tmp
                left = i;
                right=j;
    return max
```

N defa dış döngü

N-i defa iç döngü

$N+(N-1)+(N-2) \dots 1 \rightarrow \frac{N*(N+1)}{2} \rightarrow N^2$

Her durumda tüm elemanları kontrol edeceği için

$C_{best} = \Theta(n \log n)$ $C_{average} = \Theta(n \log n)$ $C_{worst} = \Theta(n \log n)$