Laborator 3
Metode de rezolvare a recurențelor de complexitate

Responsabil: Arcălianu Alexandra – arcalianu.alexandra@gmail.com

1)
   a) Scrieți un algoritm recursiv care găseşte maximul dintre elementele unui vector folosind tehnica divide et impera.
   b) Calculați complexitatea algoritmului mai întâi prin metoda iterativă şi apoi prin metoda arborelui de recurență.

2)
   a) Scrieți un algoritm recursiv care caută un element într-un vector sortat crescător folosind căutarea binară, se consideră că elementul se află în vector.
   b) Calculați complexitatea algoritmului folosind metoda substituției.

1)
   a)
```
max(v, i, j) {
        if  i = j
                return v[i]
        else{
                m = (i + j)/2
                max1 = max(v, i, m )
                max2 = max(v, m + 1,  j)
                if max1 > max2
                        return max1
                else
                        return max2
        }
}
```

   b) max(v, 1, n) $\rightarrow$ T(n)= $\begin{cases} k_{1,} n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + k_{2,} n > 1 \end{cases}$

Metoda iterativa

$$T(n) = 2T\left(n/2\right) + \Theta(1),\ T(1) = \Theta(1) \qquad |\ 2^0$$

$$T\left(n/2\right) = 2T\left(n/2^2\right) + \Theta(1) \qquad |\ 2^1$$

$$T\left(n/2^2\right) = 2T\left(n/2^3\right) + \Theta(1) \qquad |\ 2^2$$

$$\vdots$$

$$T\left(n/2^k\right) = 2T\underbrace{\left(n/2^{k+1}\right)}_{1} + \Theta(1) \qquad |\ 2^k$$

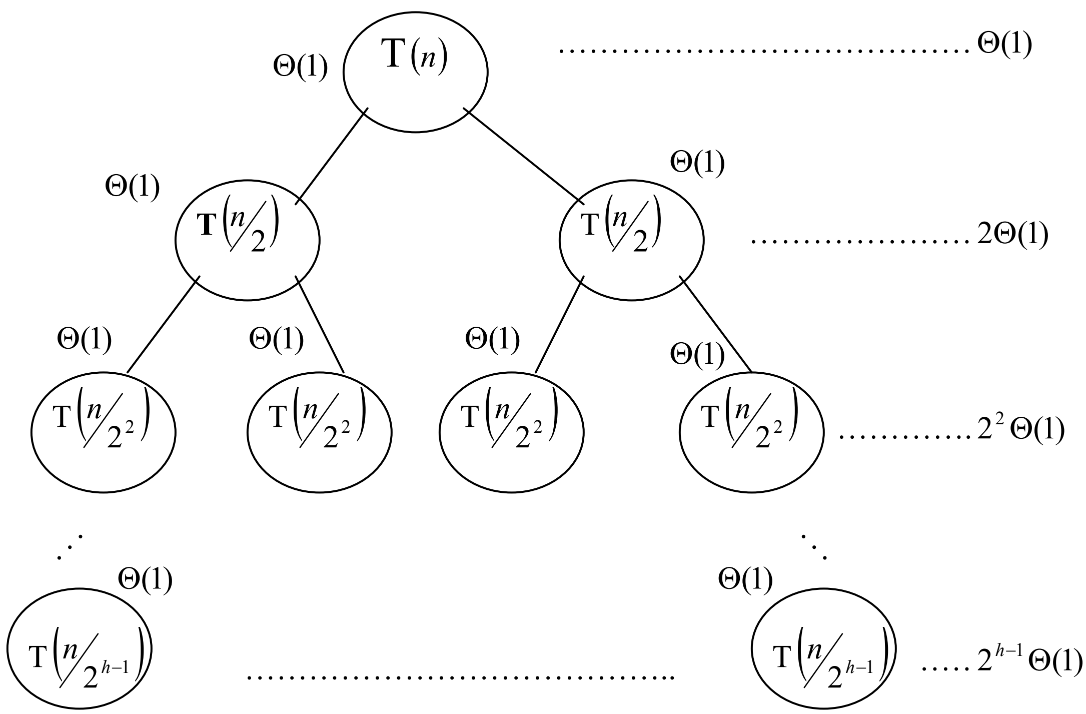
Cum n = 2^{k+1} $\Rightarrow$ k+1 = lg n

Adunând ecuațiile de mai sus rezultă:

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

Metoda arborelui de recurență

$T(n) = 2T\left(n/2\right) + \Theta(1)$, $T(1) = \Theta(1)$



Cum $n = 2^{h-1} \Rightarrow h-1 = \lg n \Rightarrow h = 1 + \lg n$

Din arborele de mai sus rezultă:

$$T(n) = \sum_{i=0}^{h-1} 2^i \Theta(1) = \Theta(1)\left(2^h - 1\right) = \Theta(1)\left(2n-1\right) = \Theta\left(2n-1\right)$$

2)

a) cautare(v, i, j, x) {
   if i=j
      return v[i]
   else {
      $m = \frac{i+j}{2}$
      if v[i] $\leq$ v[m]
         return cautare(v, i, m, x)

```
            else
                    return cautare(v, m+1, j, x)
        }
}
```

b)

cautare(v, 1, n, x) $\rightarrow$ T(n)= $\begin{cases} k_{1,} n = 1 \\ T\left(\left\lceil \frac{n}{2} \right\rceil\right) + k_{2,} n > 1 \end{cases}$

Calculăm complexitatea mai întai cu metoda iterativă şi demonstrăm complexitatea şi cu metoda substituției.

$$T(n) = T\left(n/2\right) + \Theta(1),\ T(1) = \Theta(1)$$
$$T\left(n/2\right) = T\left(n/2^2\right) + \Theta(1)$$
$$T\left(n/2^2\right) = T\left(n/2^3\right) + \Theta(1)$$
$$\vdots$$
$$T\left(n/2^k\right) = T\underbrace{\left(n/2^{k+1}\right)}_{1} + \Theta(1)$$

Cum $n = 2^{k+1} \Rightarrow k+1 = \lg n$

Din ecuatiile de mai sus rezultă:

$$T(n) = T(1) + (k+1)\ \Theta(1) = \Theta(1) + (k+1)\ \Theta(1) = (k+2)\,\Theta(1) =$$
$$= (1+\lg n)\ \Theta(1) = \Theta(1+\lg n)$$

Metoda substituției

Alegem $T(n) = T\left(\frac{n}{2}\right) + k_2,\ T(1) = k_1$

$T(n) = \Theta(1 + \lg n) \Rightarrow \exists\ c_1, c_2 \in \Re_+^*,\ \exists n_0 \in N^*$ a.î. $c_1(1 + \lg n) \le T(n) \le c_2(1 + \lg n),\ \forall n \ge n_0$

<u>Caz de bază:</u>

$n = 1 \Rightarrow c_1 \le k_1 \le c_2 \Rightarrow \boxed{n_0 = 1}$ caz de bază al inducției.

<u>Pas de inducție:</u>

Ipoteza de inducție: $T\left(\frac{n}{2}\right): c_1\left(1+\lg\frac{n}{2}\right) \le T\left(\frac{n}{2}\right) \le c_2\left(1+\lg\frac{n}{2}\right)$ (*)

Arătăm că: $T(n) \Rightarrow c_1(1 + \lg n) \le T(n) \le c_2(1 + \lg n)$ (**)

$(*) \Rightarrow c_1(1 + \lg n - 1) + k_2 \le T(n) \le c_2(1 + \lg n - 1) + k_2 \Rightarrow c_1(1 + \lg n)$
$+ (k_2 - c_1) \le T(n) \le c_2(1 + \lg n) + (k_2 - c_2) \le c_2(1 + \lg n)$

Cum trebuia să demonstrăm (**) avem :

$$\left.\begin{array}{l} \begin{cases} k_2 - c_1 \ge 0 \\ k_2 - c_2 \le 0 \end{cases} \Rightarrow \begin{cases} c_1 \le k_2 \\ c_2 \ge k_2 \end{cases} \qquad \Rightarrow c_1 \le k_1 \le c_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{Caz de bază} \end{array}\right| \Rightarrow \begin{cases} c_1 \le k_2 \le c_2 \\ c_1 \le k_1 \le c_2 \end{cases}$$

$\Rightarrow \exists c_1 = \min(k_1, k_2), c_2 = \max(k_1, k_2)$, $\exists n_0 = 1$ a.î.
$c_1(1 + \lg n) \le T(n) \le c_2(1 + \lg n)$, $\forall n \ge n_0$ (q.e.d.).