Fundamentele programării

f-8. $T(n) = 2n \in \Theta(n)$

f-9. $T(n) = n \cdot 2n = 2n^2 \in \Theta(n^2)$

$$\sum_{i=0}^{n-1}\Big(\underbrace{\sum_{j=0}^{n-1} 1}_{n} + \underbrace{\sum_{k=0}^{n-1} 1}_{n}\Big) = \sum_{i=0}^{n-1} n+n = 2n\sum_{i=0}^{n-1} 1 = 2n \cdot n$$

f-11. $\Theta(n^2 \cdot \log_{10} n)$

BC

WC

AC $\sum_{i\in D} \underbrace{P(i)}_{\text{prob}} \cdot \underbrace{E(i)}_{\text{pași pt o anumită intrare}}$

$\frac{1}{n}$ probabilitatea să găsesc un elem pe o poz

$\frac{1}{n}$

```
def search(lst, el):
    for x in lst
        if x == el
            return True
    return False
```

BC = elem pe prima poz $\Theta(1)$

WC = $el \notin lst$ $\Theta(n)$

$$AC = \sum_{\substack{i\in D\\ \text{toate intrările pos}}} P(i)E(i) = \overbrace{1\cdot\frac{1}{n}}^{\text{el pe prima poz}} + \overbrace{\underbrace{2}_{\text{2 pași}}\cdot\frac{1}{n}}^{\text{el a doua poz}} + 3\cdot\frac{1}{n} + \dots + n\cdot\frac{1}{n} =$$

$$= \frac{1+2+3+\dots+n}{n} = \frac{n(n+1)}{2n} = \frac{n+1}{2} \in \Theta(n)$$

OVERALL COMPLEXITY: $O(n)$

```
recursive_f_1
  if n <= 0
    return 1
  else
    return 1 + recursive_f_1(n-1)
```

f_1(5)
  f_1(4)
    f_1(3)
      f_1(2)
        f_1(1)
          f_1(0)

$$T(n) = \begin{cases} 1 & \text{dacă } n <= n \\ 1 + T(n-1) & \text{altfel} \end{cases}$$

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

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

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

$$\ldots$$

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

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

$$T(n) = \underbrace{1 + 1 + 1 + \ldots + 1}_{n+1} = n + 1 \in \Theta(n)$$

---

recursive_f_2

$$T(n) = \begin{cases} 1 & \text{dacă } n \leq 1 \\ 1 + T(n-5) & \text{altfel} \end{cases}$$

$T(n) = T(n-5) + 1$

$T(n-5) = T(n-10) + 1$

$\vdots$

$T(K) = 1$
$K \leq 1$

$$T(n) = \underbrace{1 + 1 + 1 + \ldots + 1}_{n/5 + 1} \in \Theta(n)$$

---

recursive_f_3

$$T(n) = \begin{cases} 1 & \text{dacă } n \leq 0 \\ 1 + T(n/2) & \text{altfel} \end{cases}$$

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

$$T(n/2) = T(n/4) + 1$$

$$T(n/4) = T(n/8) + 1$$

$$\dots$$

$$\Theta(\log_2 n)$$

---

recursive_f_5

$$T(n) = \begin{cases} 1 & \text{dacă } n \le 0 \\ 2T(n-1) + 1 & \text{altfel} \end{cases}$$

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

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

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

$$\vdots$$

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

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

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

$$\vdots$$

$$= 2^k[2T(n-k) + 1] + 2^{k-1} + 2^{k-2} + \dots + 1$$

$$k = n$$

$$2^n \overset{=1}{T(0)} + 2^{n-1} + \dots + 1$$

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

def recursiva_f_s

$$T(n) = \begin{cases} 1 & \text{dacă } n \le 1 \\ n + T(n-1) & \text{altfel} \end{cases}$$

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

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

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

$$\vdots$$

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

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

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

$$T(n) = 1 + 1 + 2 + 3 + \dots + n = 1 + \frac{n(n+1)}{2} \in \Theta(n^2)$$

recursive _ 8 _ 6

$$T(n) = \begin{cases} 1 & \text{dacă } n \le 1 \\ 4T(n/2) + 1 & \text{altfel} \end{cases}$$

$T(n) = 4T(n/2) + 1$

$T(n/2) = 4T(n/4) + 1$

$T(n/4) = 4T(n/8) + 1$

$T(n) = 4T(n/2) + 1$

$= 4(4T(n/4) + 1) + 1$

$= 4^2 T(n/4) + 4 + 1$

$= 4^2 [4T(n/8) + 1] + 4 + 1$

$= 4^3 T(n/8) + 4^2 + 4 + 1$

...

$= 4^k T\left(\frac{n}{2^k}\right) + 4^{k-1} + \ldots + 4 + 1$

pt $n = 2^k$ $\quad T(n) = 4^k T(1) + 4^{k-1} + \ldots + 4 + 1 = \frac{4^{k+1} - 1}{3}$

$$\lambda^n + \lambda^{n-1} + \lambda^{n-2} + \ldots + \lambda + 1 =$$

$$= \frac{\lambda^{n+1} - 1}{\lambda - 1}$$

f_12

complexitatea timp : $O(n \log n)$

EXTRA-SPACE COMPLEXITY $\Theta(1)$

f_13

COMPLEXITATEA TIMP : $T(n) = \begin{cases} 1 & \text{dacă } n \leq 1 \\ 2T(n/2)+1 & \text{altfel} \end{cases}$

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

$T(n/2) = 2T(n/4)+1$

$T(n/4) = 2T(n/8)+1$

$\vdots$

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

$= 2[2T(n/4)+1]+1$

$= 2^2T(n/4)+2+1$

$= 2^2[2T(n/8)+1]+2+1$

$= 2^3T(n/8)+2^2+2+1$

$\dots$

$= 2^k T(n/2^k) + 2^{k-1} + \dots + 1$

$n = 2^k$

$2^k \underbrace{T(1)}_{=1} + 2^{k-1} + \dots 1$

$= 2^k + 2^{k-1} + \dots + 1 =$

$= 2^{k+1} - 1 = 2 \cdot \underbrace{2^k}_{n} - 1 = 2n - 1 \in \Theta(n)$

SPACE COMPLEXITY

$$T(n) = \begin{cases} 1 & \text{Case } n \le 1 \\ 2T(n/2) + n & \end{cases}$$

slicing → list of copies

$T(n) = 2T(n/2) + n$

$T(n/2) = 2T(n/4) + n/2$

$T(n/4) = 2T(n/8) + n/4$

$\vdots$

$T(n) = 2T(n/2) + n$

$= 2[2T(n/4) + n/2] + n$

$= 2^2 T(n/4) + n + n$

$= 2^2 [2T(n/8) + n/4] + n + n$

$= 2^3 T(n/8) + n + n + n$

$\vdots$

$2^k T(n/2^k) + k \cdot n$

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

$\underbrace{2^k}_{n} T(1) + k \cdot n$

$n + k \cdot n =$

$= n + \log_2 n \cdot n$

$\in \Theta(n \log n)$