= COMPLEXITĂȚI =

1) "O": SPUNEM CĂ $f \in O(g)$ DACĂ $(\exists)$ c, n_0 (CONSTANTE) >0 a.î. $(\forall)$ $n \geq n_0$ AVEM $f(n) \leq c \cdot g(n)$. (FUNCȚIA SĂ FIE MĂRGINITĂ SUPERIOR DE LA UN ANUMIT RANG ÎNCOLO).

Ex: $3n^3 \in O(n^3)$; $n^2 \in O(n^3)$; $\log(n)+3 \in O(n)$ $\boxed{O: \text{"}\leq\text{"}}$

2) "Ω": SPUNEM CĂ $f \in \Omega(g)$ DACĂ $(\exists)$ c, n_0 (CONST.) >0 a.î. $(\forall)$ $n \geq n_0$ AVEM $f(n) \geq c \cdot g(n)$. (MĂRGINITĂ INFERIOR)

Ex: $n^3 \in \Omega(n^3)$; $\frac{n^3}{2} \in \Omega(n^3)$; $n^2 \in \Omega(n^3)$; $n^2-4n+17 \notin \Omega(n^3)$

$\boxed{\Omega: \text{"}\geq\text{"}}$

3) "θ": SPUNEM CĂ $f \in \theta(g)$ DACĂ $(\exists)$ $c1, c2, n_0$ (CONST.) >0 a.î. $(\forall)$ $n \geq n_0$ AVEM $c1 \cdot g(n) \leq f(n) \leq c2 \cdot g(n)$. $\boxed{\theta: \text{"}=\text{"}}$

Ex: $n^2 - 2n \cdot \log(0,5n) \in \theta(n^2)$

4) "o": SPUNEM CĂ $f \in o(g)$ DACĂ $(\forall)$ $c>0$, $(\exists)$ n_0 a.î. $(\forall)$ $n \geq n_0$ AVEM $f(n) < c \cdot g(n)$! $\boxed{o: \text{"}<\text{"}}$ → MĂRGINITĂ STRICT SUPERIOR

Ex: $n^3 \notin o(n^3)$ $\quad n^2 \in o(n^3)$

$\frac{n^3}{2} \notin o(n^3)$

5) "ω": SPUNEM CĂ $f \in \omega(g)$ DACĂ $(\forall)$ $c>0$, $(\exists)$ n_0 a.î. $n \geq n_0$ AVEM $f(n) > c \cdot g(n)$. $\boxed{\omega: \text{"}>\text{"}}$ → MĂRGINITĂ STRICT INFERIOR

Ex: $n^3 \notin \omega(n^3)$; $n^4 \notin \omega(n^4)$; $n^4 \in \omega(n^3)$; $n \cdot \log n \in \omega(n)$

$\begin{cases} O(g): \text{"}\leq\text{"} \\ o(g): \text{"}<\text{"} \end{cases}$ $\quad \theta(g): \text{"}=\text{"}$ $\quad \begin{cases} \Omega(g): \text{"}\geq\text{"} \\ \omega(g): \text{"}>\text{"} \end{cases}$

- * ÎN $\Omega(g)$ SE AFLĂ TOATE FUNCȚIILE CU TERMENUL DOMINANT "$\geq$" CEL DIN "g".
- * ÎN $O(g)$ " ——— " ——— " "$\leq$" " — " — ".
- * ÎN $\theta(g)$ " ——— " ——— " "$=$" " — " — ".

Ex:

$O(n^3)$	$\theta(n^3)$	$\Omega(n^3)$
n^2	n^3	n^4
$3n^2+n\cdot\log(n^4)$	$3n^3+n+\log n$	2^n-n^4
$11n$	$n^3+\log(n^{99})$	$4n^3+4n^2$
$n\sqrt{n}$		$n^3\sqrt{n}$

CRESTERE ASIMPTOTICĂ:

$$1 < \log n < n < n \cdot \log n < n^2 < n^2 \log n < \ldots < 2^n$$

= RECURENTE =

ÎN GENERAL, PT. UN ALGORITM RECURSIV, AVEM URMĂTORUL TIMP DE RULARE:

T(n) = T(SUBPROBLEMĂ 1) +
T(" — " — " 2) +
⋮
T(" — " — " n) + $\underbrace{D(n)}_{\text{TIMP DIVIZIUNE}} + \underbrace{C(n)}_{\text{TIMP COMBINARE}}$

❗ $T(n) = 2T\left(\frac{n}{2}\right) + 1 \Rightarrow O(n)$

❗ $T(n) = T\left(\frac{n}{a}\right) + T\left(\frac{(a-1)n}{a}\right) + O(n) \rightarrow O(n \cdot \log n)$; $T(n) = T\left(\frac{n}{b}\right) + 1 \Rightarrow \Theta(n \cdot \log n)$

MERGE SORT: $T(n) = 2T\left(\frac{n}{2}\right) + O(n) \rightarrow O(n \cdot \log n)$

INSERTION SORT: $T(n) = T(n-1) + n \rightarrow O(n^2)$

CĂUTARE BINARĂ: $T(n) = T\left(\frac{n}{2}\right) + 1 \rightarrow O(\log n)$

= TEOREMA MASTER =

APLIC PE RECURENTE DE FORMA $\boxed{T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)}$.

1) DACĂ $f(n) \in O(n^{\log_b a - \varepsilon})$, $\varepsilon > 0 \Rightarrow T(n) = \Theta(n^{\log_b a})$

2) DACĂ $f(n) \in \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \cdot \log_2 n)$

3) DACĂ $f(n) \in \Omega(n^{\log_b a + \varepsilon})$, pt $\varepsilon > 0$, $\Rightarrow T(n) \in \Theta(f(n))$

❗ $1 + \frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{n} \approx \ln n$

EXERCIȚII:

1) $n^3 \in ? \Rightarrow n^3 \in O(n^3)$

$g(n) = n^3$

$f \in O(g) \Leftrightarrow (\exists)\ c, n_0 > 0$ a.î. $(\forall)\ n \geq n_0,\ f(n) \leq c \cdot g(n)$

ALEGEM $c = 1;\ n_0 = 1 \Rightarrow n^3 \leq 1 \cdot n^3$ (A), $(\forall)\ n \geq 1$

2) $200n^3 \in O(n^3)$

$g(n) = n^3;\ f(n) = 200n^3$

$f \in O(g) \Leftrightarrow (\exists)\ c, n_0 > 0$ a.î. $(\forall)\ n \geq n_0,\ f(n) \leq c \cdot g(n)$

ALEGEM $c = 201,\ n_0 = 1 \Rightarrow 200n^3 \leq 201 \cdot n^3$ (A), $(\forall)\ n \geq 1$

*3) $n \cdot \log_2 n \in \Theta(\log_2 n!) \Leftrightarrow \log_2 n! \in \Theta(n \cdot \log_2 n)$

$f(n) = n \cdot \log_2 n$

$g(n) = \log_2 n!$

$f \in \Theta(g) \Leftrightarrow (\exists)\ c_1, c_2, n_0 > 0$ a.î. $(\forall)\ n \geq n_0,\ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

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

$c_1 \cdot \log_2 n! \leq n \cdot \log_2 n = \log_2 n^n$

Alegem $c_1 = 1,\ n_0 = 1 \Rightarrow \log_2 n! \leq \log_2 n^n \Leftrightarrow n! \leq n^n$ (A)

DEM. PRIN INDUCȚIE;

$n = 1 \Rightarrow 1! = 1 \leq 1^1$ (A)

$P(k) \rightarrow P(k+1)$

$P(k):\ k! \leq k^k$ A

$P(k+1):\ (k+1)! \leq (k+1)^{k+1}$

$k!\cancel{(k+1)} \leq (k+1)^k \cdot \cancel{(k+1)}$

$k! \leq (k+1)^k$ (A) $(k! \leq k^k)$

b) $f(n) \leq c_2 \cdot g(n)$

$\log_2 n^n \leq c_2 \cdot \log_2 n!$

$\log_2 n! = \log_2 1 + \ldots + \log_2 n \geq \log_2 \frac{n}{2} + \ldots + \log_2 n \geq \frac{n}{2} \log_2(n/2) =$

$= \frac{n}{2}(\log_2 n - 1) \geq c_2 \cdot n \cdot \log_2 n$

$c_2 = \frac{1}{4};\ n_0 = 4$

4) $f(n) + g(n) \in \Theta(\max\{f(n); g(n)\})$

$(\exists)\ c_1, c_2, n_0 > 0$ a.î. $(\forall)\ n \geq n_0 \Rightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \to$ CAZ GENERAL

a) $\left.\begin{array}{l} c_1 \cdot \max\{f(n); g(n)\} \leq f(n) + g(n) \\ c_1 = 1;\ n_0 = 1 \end{array}\right\} \Rightarrow A$

b) $\left.\begin{array}{l} f(n) + g(n) \leq c_2 \cdot \max\{f(n), g(n)\} \\ c_2 = 2;\ n_0 = 1 \end{array}\right\} \Rightarrow A$

5) $\left.\begin{array}{l} f \in O(g) \\ g \in O(h) \end{array}\right\} \Rightarrow f \in O(h)$

$\left.\begin{array}{l} f \in O(g) \Leftrightarrow (\exists)\ c_1, n_0 > 0 \text{ a.î. } (\forall)\ n \geq n_0,\ f(n) \leq c_1 \cdot g(n) \\ g \in O(h) \Leftrightarrow (\exists)\ c_2, n_0' > 0 \text{ "} \text{———} \text{"} \text{———} \text{"};\ g(n) \leq c_2 \cdot h(n) \end{array}\right\} \Rightarrow$

$\left.\begin{array}{l} \Rightarrow f(n) \leq c_1 \cdot g(n) \leq c_1 \cdot c_2 \cdot h(n) \Rightarrow f(n) \leq c_1 \cdot c_2 \cdot h(n) \\ f \in O(h) \Leftrightarrow (\exists)\ c, n_0 > 0 \text{ a.î. } (\forall)\ n \geq n_0,\ f(n) \leq c \cdot h(n) \end{array}\right\} \Rightarrow f \in O(h)$

6) $o(f(n)) \cap \omega(f(n)) = \emptyset$

$g \in o(f(n)) \Leftrightarrow (\forall)\ c_1 > 0, (\exists)\ n_0$ a.î. $(\forall)\ n > n_0;\ g(n) < c_1 \cdot f(n)$

$g \in \omega(f(n)) \Leftrightarrow (\forall)\ c_2 > 0, (\exists)$ " ——— " ——— "; $g(n) > c_2 \cdot f(n)$

PRESUPUN PRIN ABSURD CĂ $g(n) \in (o(f(n)) \cap \omega(f(n))) \Rightarrow$

$\Rightarrow \begin{cases} g(n) \in o(f(n)) \Rightarrow g(n) < c_1 f(n) \\ g(n) \in \omega(f(n)) \Rightarrow g(n) > c_2 f(n) \end{cases} (\forall)\ c_1, c_2$

Aleg $c_1 = c_2 = 1 \Rightarrow \begin{cases} g(n) < f(n) \\ g(n) > f(n) \end{cases};\ (\forall)\ n > n_0; n_0'$

Aleg $n'' > \max(n_0 \text{ și } n_0') \Rightarrow \left.\begin{cases} g(n'') < f(n'') \\ g(n'') > f(n'') \end{cases}\right\} \Rightarrow \not\exists$

7) $(n+a)^b \in \Theta(n^b)$; $a>0$; $b>1$

$f\in\Theta(g) \Rightarrow (\exists)\, c_1; c_2; n_0>0$ a.î. $(\forall)\, n>n_0 \Rightarrow c_1\cdot g(n) \le f(n) \le c_2\cdot g(n)$

a) $c_1\cdot n^b \le (n+a)^b$

$$\left.\begin{array}{l} c_1\cdot n^b \le n^b + C_b^1 n^{b-1}a + \ldots + C_b^b a^b \\ c_1=1 \text{ și } n_0=1 \end{array}\right\} \Rightarrow (A) \Rightarrow 0 \le C_b^1 n^{b-1}a + \ldots + C_b^b a^b$$

b) $(n+a)^b \le c_2\cdot n^b$

$$(n+a)^b = n^b + C_b^1 n^{b-1}a + \ldots + C_b^b a^b \le n^b(1 + C_b^1 a + \ldots + C_b^b a^b) \le$$

$$\le n^b\cdot a^b(1 + C_b^1 + C_b^2 + \ldots + C_b^b) = n^b\cdot a^b\cdot 2^b(b+1)$$

$c_2 = a^b\cdot 2^b(b+1)$; $n_0=1 \Rightarrow A$

8) $f(n) + o(f(n)) \in \Theta(f(n))$

$f\in\Theta(g) \Leftrightarrow (\exists)\, c_1, c_2, n_0>0$ a.Î. $(\forall)\, n>n_0 \Rightarrow c_1\cdot g(n) \le f(n) \le c_2\cdot g(n)$

a) $c_1\cdot f(n) \le f_n + o(f(n))$

$c_1=1 \Rightarrow A$, $(\forall)\, n>n_0$

b) $f(n) + o(f(n)) \le c_2\cdot f(n)$

$g\in o(f(n)) \Rightarrow (\forall)\, c>0, (\exists)\, n_0$ a.î. $(\forall)\, n>n_0 \Rightarrow g(n) < c\cdot f(n)$

$$\left.\begin{array}{l} f(n) + o(f(n)) < f(n) + c\cdot f(n) \le 2\cdot f(n) \\ c_2 = 2 \end{array}\right\} \Rightarrow A$$

---

1) $T(n) = T\left(\frac{n}{a}\right) + T\left(\frac{(a-1)n}{a}\right) + \overset{O(n)}{\overset{\parallel}{n}} \in O(n\cdot\log n)$ → MERGESORT

$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$

PRESUPUN CĂ $T\left(\frac{n}{3}\right) \le c\cdot\frac{n}{3}\cdot\log_2\frac{n}{3}$

TB. SĂ ARĂTĂM CĂ $T(n) \le c\cdot n\cdot\log_2 n$

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n \le c\cdot\frac{n}{3}\cdot\log_2\frac{n}{3} + 2c\cdot\frac{n}{3}\cdot\log_2\frac{2n}{3} + n =$$

$$= c\cdot\frac{n}{3}\left(\log_2 n - \log_2 3 + 2\cdot(\underbrace{\log_2 2}_{1} + \log_2 n - \log_2 3)\right) + n =$$

$$= c\cdot\frac{n}{3}(3\cdot\log_2 n - 3\cdot\log_2 3 + 2) + n = cn\left(\log_2 n - \log_2 3 + \frac{2}{3}\right) + n =$$

$$= c\cdot n\cdot\log_2 n - cn\underbrace{\left(\log_2 3 - \frac{2}{3} - \frac{1}{c}\right)}_{>0} \le c\cdot n\cdot\log_2 n \ \checkmark$$

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

PRESUPUN $T\left(\frac{n}{2}\right) \leq c \cdot \frac{n}{2} - b$

TB. SĂ DEM. $T(n) \leq c \cdot n - b$

$$T(n) = 2T\left(\frac{n}{2}\right) + 1 \leq \cancel{2}c \cdot \frac{n}{\cancel{2}} - 2b + 1 = cn - 2b + 1 \underset{b>1}{\leq} cn - b \Big\} \Rightarrow \checkmark$$

3) $T(n) = T(n/2) + n \in O(n)$

PRESUPUN $T(n/2) \leq c \cdot \frac{n}{2}$

TB. SĂ DEM. $T(n) \leq c \cdot n$

$$T(n) = T(n/2) + n \leq c \cdot \frac{n}{2} + n = n\left(1 + \frac{c}{2}\right) \leq c \cdot n, \ (\forall)\ c \geq 2$$

4) $T(n) = T(n-1) + 1 \in O(n)$

PRESUPUN $T(n-1) \leq c(n-1)$

TB. SĂ DEM. $T(n) \leq c \cdot n$

$$T(n) = T(n-1) + 1 \leq c(n-1) + 1 = cn - (c-1) \leq cn, \ (\forall)\ c \geq 1$$

5) $T(n) = T(n-1) + \overset{O(n)}{n} \in O(n^2)$ → INSERTION SORT

PRESUPUN $T(n-1) \leq c \cdot (n-1)^2$

TB. SĂ DEM $T(n) \leq c \cdot n^2$

$$T(n) = T(n-1) + n \leq c(n-1)^2 + n = c(n^2 - 2n + 1) + n = \cancel{c \cdot n^2} - 2 \cdot c \cdot n + c + n \leq \cancel{cn^2}$$

$$-2 \cdot c \cdot n + c + n \leq 0 \Rightarrow 2 \cdot c \cdot n - c - n \geq 0$$

$$n\left(2c - 1 - \frac{c}{n}\right) \geq 0, \ (A), \ (\forall)\ c \geq 1$$

6) $T(n) = T\left(\frac{n}{2}\right) + 1 \in O(\log n)$

PRESUPUN $T\left(\frac{n}{2}\right) \leq c \cdot \log_2 \frac{n}{2}$

TB. SĂ DEM: $T(n) \leq c \cdot \log_2 n$

$$T(n) = T\left(\frac{n}{2}\right) + 1 \leq c \cdot \log_2 \frac{n}{2} + 1 = c(\log_2 n - \overbrace{\log_2 2}^{1}) + 1 = c \cdot \log_2 n - (c-1) \leq$$

$$\leq c \cdot \log_2 n, \ (\forall)\ c \geq 1$$

7) $T(n) = 2T(n-1) + 1 \in O(2^n)$

PRESUPUN $T(n-1) \leq c \cdot 2^{n-1} - b$

TB. SĂ DEM $T(n) \leq c \cdot 2^n - b$

$$T(n) = 2T(n-1) + 1 \leq 2 \cdot c \cdot 2^{n-1} - 2b + 1 = c \cdot 2^n - 2b + 1 \leq c \cdot 2^n - b, \ (\forall)\ b \geq 1$$

= SORTĂRI =

$O(n^2)$: BUBBLE / INSERTION / SELECTION SORT

$O(n \cdot \log n)$: MERGE / QUICK / HEAP SORT

$O(n)$: COUNT / RADIX SORT

---

1) BUBBLESORT:

5 1 4 2 8 9 (LUĂM GRUPURI DE 2 ELEMENTE)

1<5 => ÎNTERSCHIMB

1 5 4 2 8 9

4<5 => SWAP

1 4 5 2 8 9

2<5 => SWAP

1 4 2 5 8 9

PARCURGI LISTA PÂNĂ CÂND E ORDONATĂ. IEI "ok=0" CA SĂ VERIFICI DACĂ S-AU PRODUS ÎNTERSCHIMBĂRI. ok=1 => STOP.

1 2 4 5 8 9 → SORTAT

2) INSERTION SORT:

SORTAT | ZONA NESORTATĂ

7 | 8 5 2 4 6 3

7<8 → NU SE ÎNTERSCHIMBĂ

7 8 | 5 2 4 6 3

8>5; 7>5

5 7 8 | 2 4 6 3

2<8; 2<7; 2<5

2 5 7 8 | 4 6 3

4<8; 4<7; 4<5; 4>2

2 4 5 7 8 | 6 3

6<8; 6<7; 6>5

2 4 5 6 7 8 | 3

3<8 .... 3<4; 3>2

2 3 4 5 6 7 8 → SORTAT

3) SELECTION SORT:

SORTAT | NESORTAT

| (64) 25 12 22 11

SWAP

PARCURGEM VECTORUL ȘI GĂSIM MINIMUL.

11 | (25) 12 22 64

11 12 | (25) 22 64

11 12 22 | (25) 64

11 12 22 25 64 → SORTAT

---

4) MERGESORT: (DIVIDE ET IMPERA) ⇒ $T(n) = 2T(n/2) + O(n) \in O(n \cdot \log n)$

38 27 43 3 | 9 82 10

38 27 | 43 3 9 82 | 10

38 | 27 43 | 3 9 | 82 10

38 > 27 43 > 3 9 < 82 10

DIVIDE

27 38 3 43 9 82 10

3 27 38 43 9 10 82

(27 > 3; 27 < 43; 38 < 43) (9 < 10; 82 > 10)

3 9 10 27 38 43 82 ⇒ SORTAT

(3 < 9; 27 > 9; 27 > 10; 27 < 82 ....)

MERGE

5) QUICKSORT: $\Rightarrow T(n) = T(\frac{n}{a}) + T(\frac{(a-1)n}{a}) + O(n) \in O(n \cdot \log n)$

DIVIDE ET IMPERA

PIVOT (TOATE EL. DIN STÂNGA $\Rightarrow$ "$\leq$"; DIN DREAPTA $\Rightarrow$ "$>$")

5 11 22 7 4 900 100 1 32 $\boxed{13}$

$i=0$, $i=1$, $i=2$; $j=0$

$5 < 13$; $11 < 13$; $22 \not< 13 \to$ SWAP (22; 13)

5 11 $\boxed{13}$ 7 4 900 100 1 32 22

$j=2$, $j=1$

$13 < 32$; $13 \not< 1 \to$ SWAP (13; 1)

5 11 1 7 4 900 100 $\boxed{13}$ 32 22

$i=3$, $i=4$, $i=5$

$7 < 13$; $4 < 13$; $900 \not< 13 \to$ SWAP (900, 13)

5 11 1 7 $\boxed{4}$ | $\boxed{13}$ | 100 900 32 $\boxed{22}$ (13 < 100 ✓)

$j=3$

APLICĂM ACEIAȘI PAȘI PT PARTEA STÂNGĂ ȘI APOI PT. PARTEA DR..

5 11 1 7 $\boxed{4}$

$i=0$, $j=0$

$5 \not< 4 \to$ SWAP

$\boxed{4}$ 11 1 7 5

$j=2$, $j=1$

$4 < 7$; $4 \not< 1 \to$ SWAP

1 11 $\boxed{4}$ 7 5

$i=1$

$11 \not< 4 \to$ SWAP

$\boxed{1}$ | $\boxed{4}$ | 11 7 $\boxed{5}$

SORTAT

$i=0$, $j=0$

$\boxed{5}$ | 7 $\boxed{11}$

100 900 32 $\boxed{22}$

$i=0$, $j=0$

$100 \not< 22 \to$ SWAP

$\boxed{22}$ | 900 32 $\boxed{100}$

$j=2$, $j=1$

$22 < 32$; $22 < 900$

900 32 $\boxed{100}$

$i=0$, $j=0$

$900 \not< 100 \to$ SWAP

$\boxed{100}$ 32 900

$j=1$

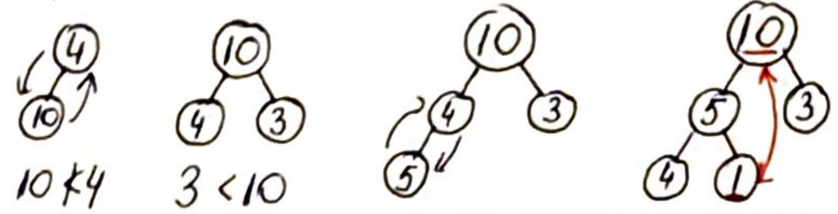
$100 \not< 32 \to$ SWAP

32 | $\boxed{100}$ | 900

1 4 5 7 11 13 22 32 100 900 $\Rightarrow$ SORTAT

6) HEAPSORT: $\in O(n \cdot \log n)$

CREĂM HEAP CU ELEMENTELE DATE: 4 10 3 5 1 (MAX HEAP)



$10 \nless 4$ $3 < 10$ $5 \nless 4$; $5 < 10$ $1 < 5$

VECTORUL: 10 5 3 4 1

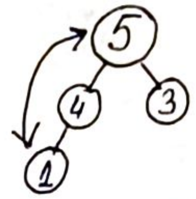
SWAP PRIMUL ȘI ULTIMUL ELEMENT DIN HEAP ȘI ELIMINĂM ULTIMUL ELEMENT

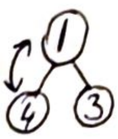
VECTOR: 1 5 3 4 | 10

VECTOR: 5 4 3 1 | 10

(REARANJĂM HEAP)

=>





VECTOR: 1 4 3 | 5 10

=>

V: 4 1 3 | 5 10

VECTOR: 3 1 | 4 5 10 =>

V: 1 | 3 4 5 10

=> V: 1 3 4 5 10 => SORTAT

7) COUNTING SORT: $\Rightarrow O(n) \Rightarrow$ { DACĂ AM UN ȘIR DE "n" ELEMENTE ÎN CARE FIECARE ELEMENT ARE MAX. "K" APARIȚII $\Rightarrow O(n+K)$

INPUT: 1 4 1 2 7 5 2 $\Rightarrow$ CHEILE $\in [0,9] \cap \mathbb{N}$

INDEX:	0	1	2	3	4	5	6	7	8	9	VECTOR FRECVENȚĂ
	0	2	2	0	1	1	0	1	0	0	$\Rightarrow$ DE CÂTE ORI APAR CHEILE

0+2 2+2 4+0 4+1 5+1 6+0 6+1 7+0 7+0

INDEX:	0	$\boxed{1}$	2	3	4	5	6	7	8	9	
	0	$\boxed{2}$	4	4	5	6	6	7	7	7	$\Rightarrow$ SUMA APARIȚII

$\hookrightarrow$ 1 apare la poziția 2; 2-1=1

AVEM 7 ELEMENTE ÎN INPUT $\Rightarrow$ VECTOR DE POZIȚII CU 7 ELEMENTE.

POZIȚII:	1	2	3	4	5	6	7	
	1	1	2	2	4	6	7	$\Rightarrow$ SORTAT

INPUT: 1 4 1 2 7 5 2 (PARCURGEM INPUT PAS CU PAS ȘI FACEM MODIFICĂRI ÎN "INDEX")

poz:2
2-1=1

poz:5 $\Rightarrow$ 5-1=4

poz:1
1-1=0

2 la poz:4
4-1=3

NU E OPTIM PENTRU VALORI MARI.

SE FOLOSEȘTE ÎMPREUNĂ CU RADIXSORT.

```
for(i=0;i<n;i++)
    frecvența[v[i]]++;
for(i=1;i<=EL_MAX;i++)
    frecvența[i]+=frecvența[i-1];
for(i=0;i<n;i++)
{   output[frecvența[v[i]]-1]=v[i];
    frecvența[v[i]]--;
}
```

8) RADIX SORT: ⇒ $O(n)$ $(O(n+k))$

INPUT: 170 45 75 90 802 24 2 66

ÎNCEPEM DE LA CEL MAI NESEMNIFICATIV BIT (CIFRA UNITĂȚILOR).

LE SORTĂM FOLOSIND COUNTING SORT. (DACĂ (∃) CIFRE EGALE, SE PĂSTREAZĂ ORDINEA ORIGINALĂ.

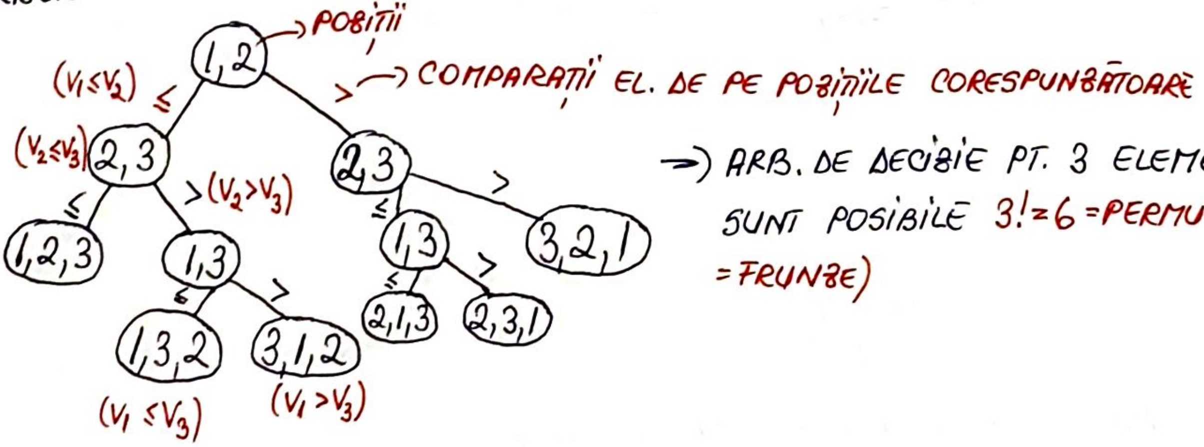
170 90 802 _2 24 45 75 66

802 __2 _24 _45 _66 170 _75 _90

2 24 45 66 75 90 170 802 ⇒ SORTAT

ORICE ALGORITM DE SORTARE BAZAT PE COMPARAȚII ÎNTRE CHEI ARE TIMP DE RULARE (ÎN CEL MAI DEFAVORABIL CAZ) = $\boxed{\Omega(n \cdot \log n)}$

ALGORITMII DE SORTARE BAZAȚI PE COMPARAȚIA DINTRE CHEI POT FI REPREZENTAȚI CA ARBORI DE DECIZIE.



⇒ ARB. DE DECIZIE PT. 3 ELEMENTE. SUNT POSIBILE $3! = 6$ = PERMUTĂRI (= FRUNZE)

ÎN CEL MAI DEFAVORABIL CAZ, NR. DE COMPARAȚII REALIZAT DE ARBORELE DE DECIZIE ESTE EGAL CU CEL MAI LUNG DRUM DE LA RĂDĂCINĂ, LA FRUNZE; ADICĂ ESTE EGAL CU ÎNĂLȚIMEA ARBORELUI.

ARB. BINAR CU ÎNĂLȚIMEA "h" NU ARE MAI MULT DE 2^h FRUNZE.

$$n! \le 2^h \mid \log_2$$

$$\log_2 n! \le \log_2 2^h = h$$

PRIN APROXIMAREA LUI STIRLING $\left(n! \ge \left(\frac{n}{e}\right)^n\right)$ ⇒ $h \ge \log\left(\frac{n}{e}\right)^n = n \cdot \log n - n \log e = \Omega(n \cdot \log n)$