Algoritmi de sortare

Analiza algoritmilor

n	log n	n	n log n	n^2	n^3	2^n	n!
10	3	10	33	10^2	10^3	10^3	10^5
10^2	7	10^2	664	10^4	10^6	10^{30}	10^{94}
10^3	10	10^3	9966	10^6	10^9	10^{301}	10^{1435}
10^4	13	10^4	132877	10^8	10^{12}	10^{3010}	10^{19335}
10^5	17	10^5	1660964	10^{10}	10^{15}	10^{30103}	10^{243338}
10^6	20	10^6	19931569	10^{12}	10^{18}	10^{301030}	$10^{2933369}$

- Complexitatea algoritmilor
- Timp minim, mediu, maxim
- Caz favorabil (cel mai bun caz), nefavorabil (cel mai rau caz)

Sortarea

- Comparison Sorting Visualization (usfca.edu)
- Data Structure Visualization (usfca.edu)

INDICII VECT sunt [1, ---, n]

Sortare prin selectie – Sortare crescătoare

n = nr. de elemente

v: 10, 4, 2, 6, 7, 8, 3, 9 → 2, 3, 4, 6, 7, 8, 9, 10

// determină cel mai mic elem din v[1], v[2], ---, v[n]

j = indicele celui mai mic

v[1] ↔ v[j]

// determină cel mai mic elem din v[2], v[3], --, v[n]

j = indicele celui mai mic

v[2] ↔ v[j]

// determină cel mai mic elem din v[n-1], v[n]

v[n-1] ↔ v[j]

j = ---

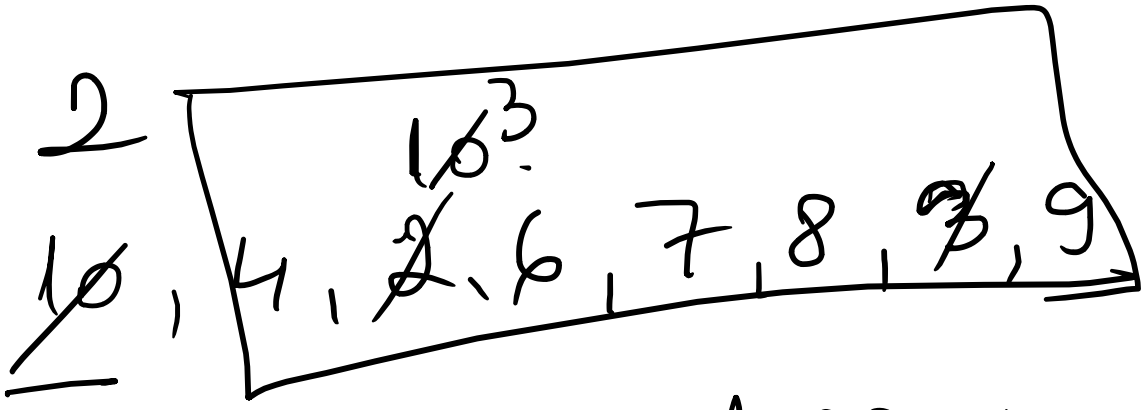
Sortare prin selectie

Op. de bază = comparația

- For i=1, n-1
- // determinam min din secventa v[i],…, v[n]. Fie j indicele lui.
- min= v[i], j = i
- for k = i+1, n
- if v[k]<min then min = v[k], j=k
- endif
- endfor
- if j != i then v[i] <-> v[j] // interschimba
- endif
- endfor
-

nr. de comp

$$1+2+\dots+K=\frac{K(K+1)}{2}$$



for $i=1$ — $n-1$ comp.

$i=2$ — $n-2$ comp

$\vdots$

$i=n-1$ — 1 comp.

total $= 1+2+3+\dots+(n-1) = \frac{(n-1)\,n}{2} = \frac{n^2-n}{2}$

$\approx \Theta(n^2)$.

$v =$ 

3 comp.
2 comp.
1 comp.

Indiferent de vectorul v, nr. de comparații este același.

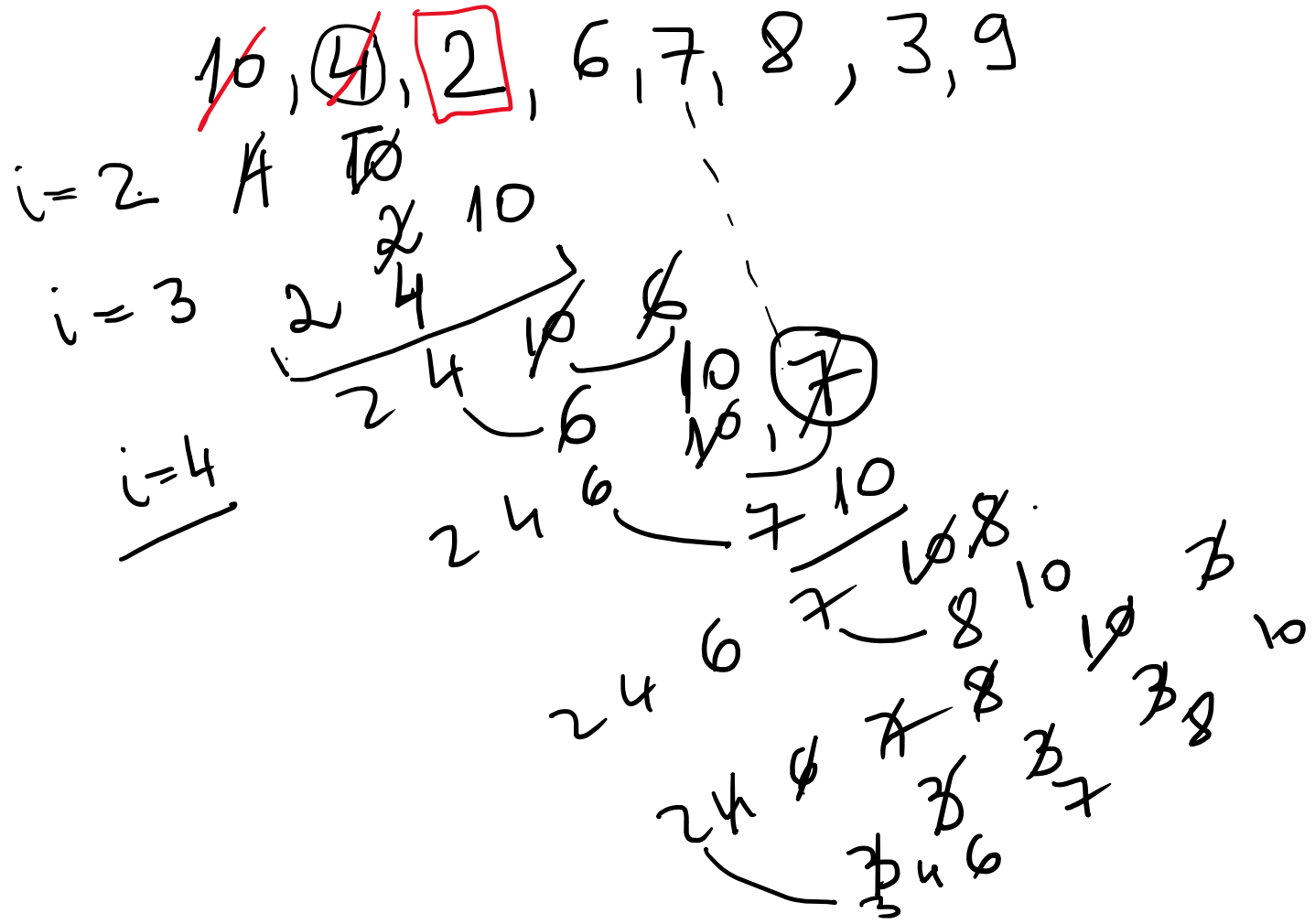
Alg este de ordin $\Theta(n^2)$.

Sortare prin inserare

Pp $v[1] < v[2] < \dots < v[i]$

vrem să inserăm $v[i+1]$

ai după $v[1] < v[2] < \dots < v[i+1]$

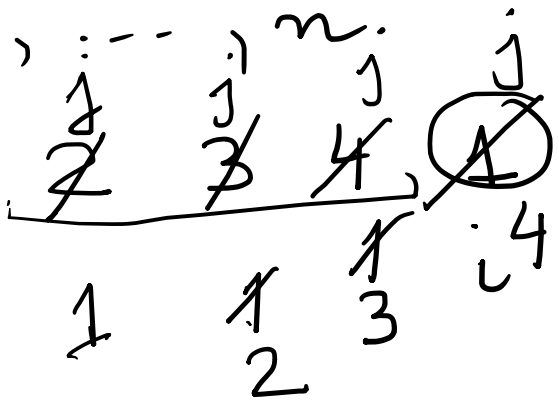


2 3 4 6 7 8 10 9

9 10

2 3 4 6 7 8 9 10.

indici 1, 2, ... i ... n.



j j
8 10
8 10

2, 4, 6, 7, 10, 8

i

```
for i=2,n
    j=i
    while(j>1 && v[j]<v[j-1])
        v[j] <-> v[j-1]
        j--
    endwhile
endfor
```

Analiza algoritmului

Nr. min de comp.

nr. comp. $= n-1$.

1, 2, 3, 5, 6, (7).

timp minim $\sim \Theta(n)$.

Nr. max de comp = șir descrescător

7, 6, 5, 2, (1).

pt v[i] se fac i-1 comp

$i = 2 \rightarrow$ 1 comp

$i = 3 \rightarrow$ 2 comp

$i = n \rightarrow n-1$ comp

total $= 1 + 2 + \dots + n-1$

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

timp min $= \Theta(n)$

timp maxim $= \Theta(n^2)$

Se poate arăta că timp mediu $\sim \Theta(n^2)$.