

Sortāri Simple

- Insertie
- Selectie
- Interschimbare

• operatii \rightarrow compare
 \downarrow interschimbare

void swap(int *a, int *b)

\downarrow int tmp = *a;

*a = *b;

\downarrow *b = tmp;

void cool_swap (- - -)

{ $(\ast a) = (\ast a) \wedge (\ast b)_i$

$(\ast b) = - - - -$

} $(\ast a) = - - - -$

$$a \wedge a = 0$$

$$a \wedge 0 = a$$

$$a \wedge b = a \oplus b =$$

$$a = a \wedge b$$

$$b = (a \wedge b) \rightarrow \underbrace{a \wedge b \wedge b}_{0} = a$$

$$a = (a \wedge b) \vdash (a \wedge b) \wedge a$$

Sortare stabilită →

→ el. cu aceeași cheie și mențin
poziția initială

Tablouri \rightarrow mem. centrală

Sortările \rightarrow mem. RAM

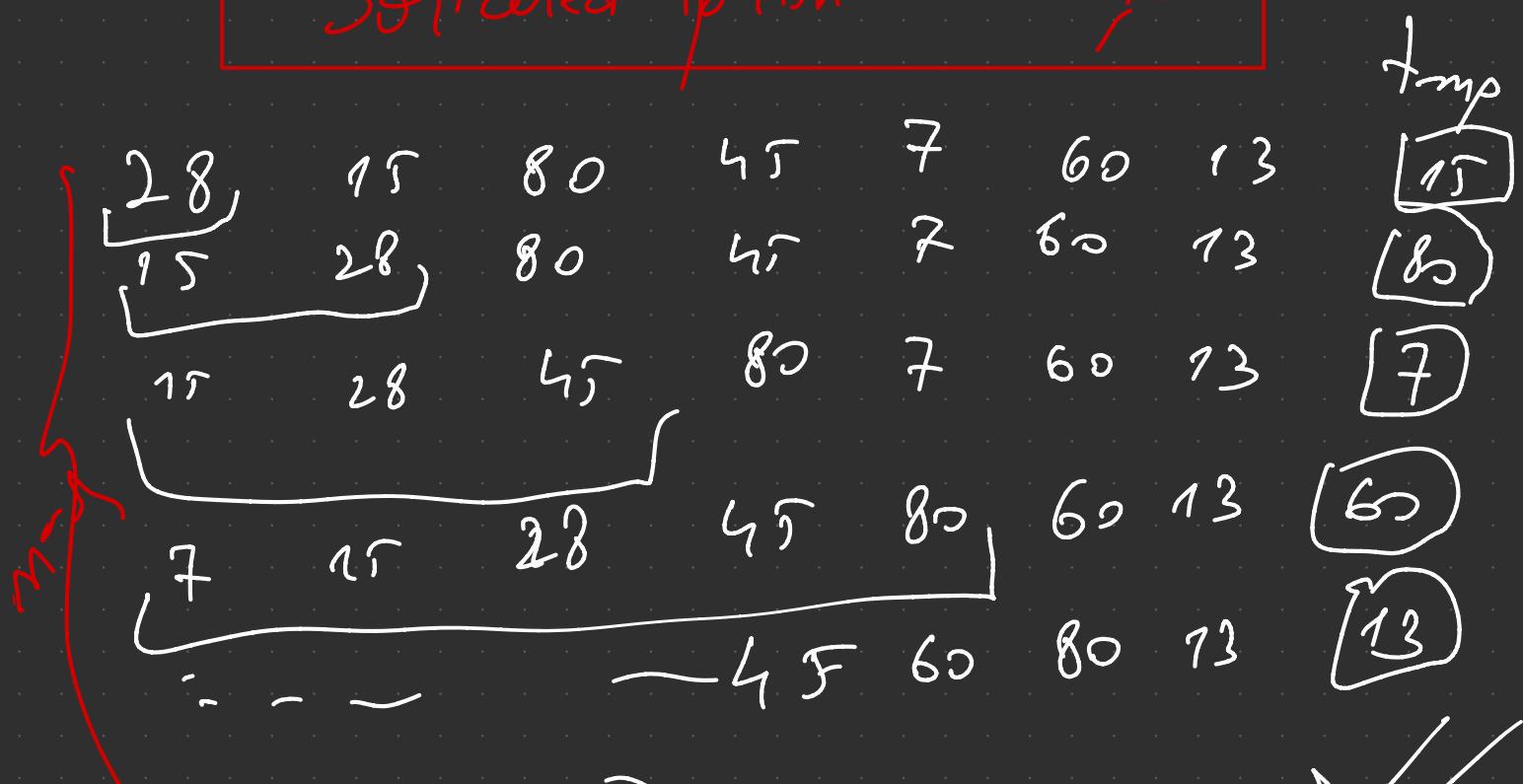
Sortarea in Situ^n

\rightarrow nu se folosesc tablouri auxiliare

Metode directe de sortare

$\rightarrow O(n^2)$

Sortarea prin inserție



$O(n^2)$

$n - 1 - i$

a, m

$a^1 a = 0$
 $a^1 0 = a$
 $= a$

i comparatii
 i mutari

$\left\{ \begin{array}{l} \text{for } (i=1; i < m-1; i++) \\ \quad t_{mp} = a[i]; \\ \text{for } (j=i; (j > 0) \& (a[j] > t_{mp}); j--) \\ \quad a[j] = a[j-1]; \\ \quad a[j] = t_{mp}; \end{array} \right.$

$$C_{\max} = \sum_{i=1}^{m-1} i = \frac{m(m-1)}{2} \Rightarrow O(n^2)$$

$$C_{\min} = (m-1) \cdot 1 \Rightarrow O(n)$$

(deja sortat)

$$C_{\text{mediu}} = O(n^2)$$

Mutari :

$$M_{\max} = \sum_{i=1}^{n-1} (i+2) \Rightarrow O(n^2)$$

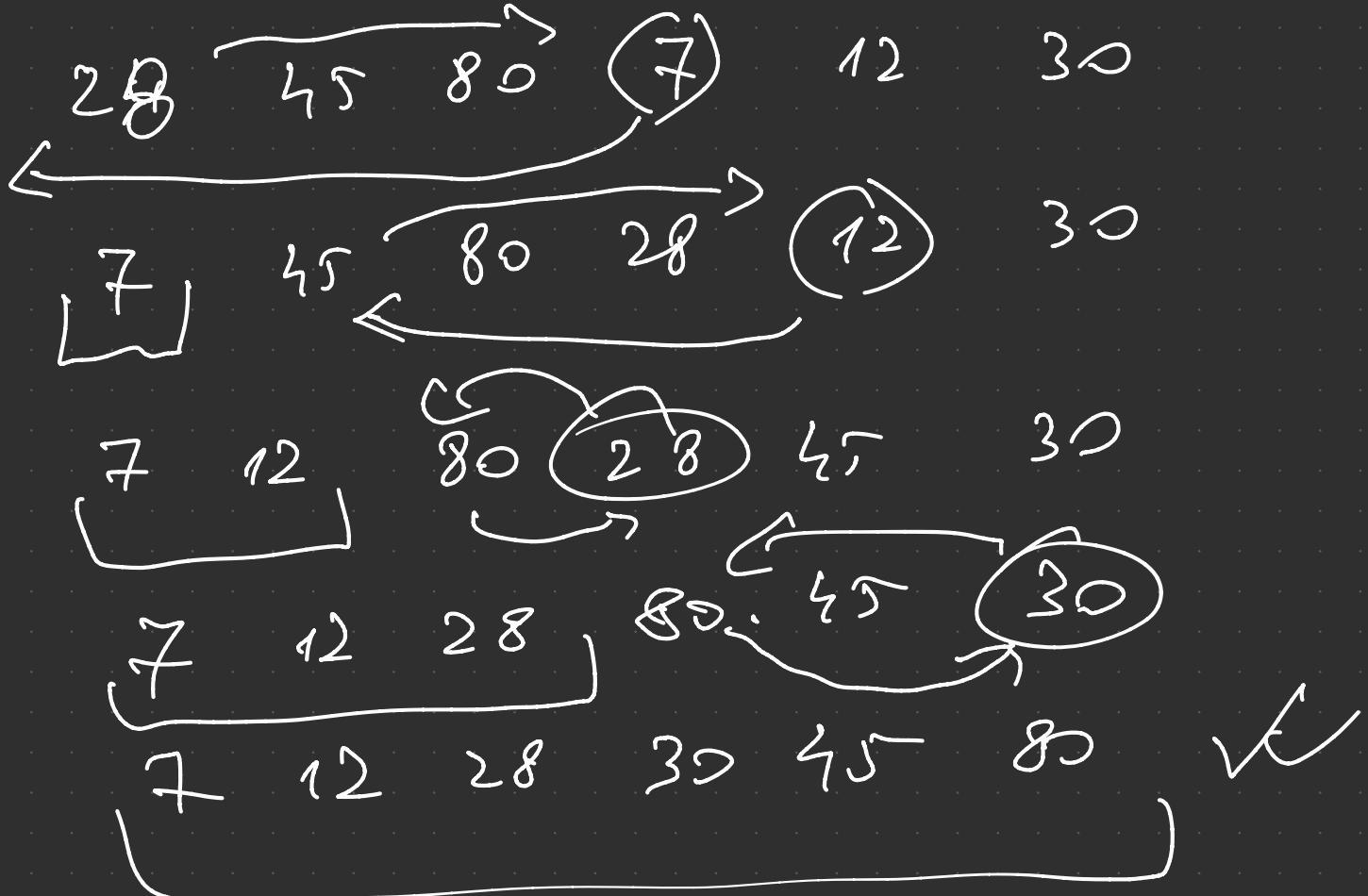
Avantajie

- simplă

Devantajie

ineficientă
(caz median = cost nefavorabil)

Sortare prin selecție



```

for (index = 0; i = 0; i < n - 1; i++) C(n-1)
    {
        index = i;
        for (j = i + 1; j < n; j++) C(m-1-i)
            if (a[j] < a[index])
                index = j;
        max(&a[1], &a[index]); M
    }
}

```

$$C_{\max} = (n-1)(n-1-i) \Rightarrow O(n^2)$$

$$\sum_{i=0}^{n-1} (n-1-i) = (n-1)n(n-2) + \dots + 2+1 = \frac{n(n-1)}{2}$$

$$M = 3(n-1) \quad \underline{\underline{O(n)}}.$$

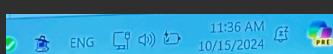
28 45 80 7 12 30
↙ ↘ ↙ ↘ ↙ ↘
swap

28 45 7 80 12 30
sum

28 45 7 12 80 30

28 45 7 12 30 80
 \ \ \ \ \ \
 3n + p

28 7 45 12 30 80



```

for ( i = 0 ; i < m ; i++ )
{
    for ( j = m-1 , j > i ; j-- )
        if ( a[ j-1 ] > a[ j ] )
            swap( a[ j-1 ], a[ j ] );
}

```