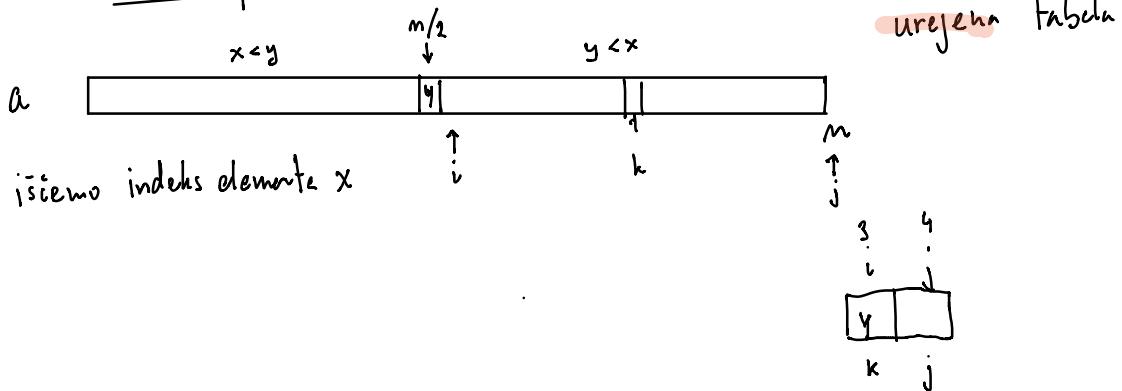


Bisekcija:



## Urejanje tabel

Nalogen: dano tabelo elementov uredi glede na dano ureditve.

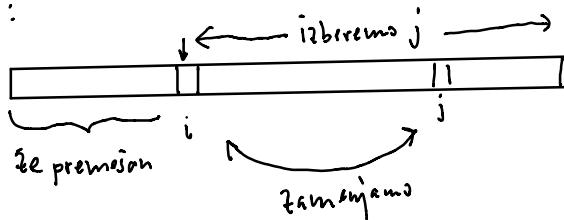
Vaja: Kako enakomerno premesimo tabelo?

$$a = [a_0, \dots, a_{n-1}]$$

$\downarrow$       nahajanje permutacije  $\pi$  na  $\{0, \dots, n-1\}$

$$[a_{\pi(0)}, \dots, a_{\pi(n-1)}]$$

Postopek:



$$\begin{array}{c} 0 \\ \hline \boxed{y} \quad z \end{array}$$

$$\frac{1}{n} \times \frac{n-1}{n} \times \frac{1}{n-1} = \frac{1}{n}$$

Dve vrsti urejanja:

- urejanje na mestu: tabelo a sprememimo, da je urejena (in place)
  - urejanje, ki prvočne tabele ne spremeni, vrne novo urejeno tabelo

## Urejanje na mestu, pri poskuš:

The diagram illustrates the selection sort algorithm on an array of numbers. The array is shown in three rows:

- Initial Values:** An array of 10 elements labeled  $a_0, a_1, a_2, a_3, \dots, a_{n-2}, a_{n-1}$ . The values are 3, 1, 5, 8, 2, 4, 7, 1, 6.
- Intermediate State:** The array after one pass of the algorithm. The element  $a_8$  has been moved to its correct position at index 8. Indices  $i$  and  $j$  are used to track the current minimum search range.  $i$  starts at 0 and  $j$  starts at 1. Arrows show the swap of  $a_8$  and  $a_j$ .
- Final Sorted State:** The array is fully sorted: 1, 2, 5, 8, 3, 4, 7, 1, 6.

To je urejanje z izbiranjem:

Vhod: tabela  $a = [a_0, \dots, a_{n-1}]$ ,  $n = \text{len}(a)$

for  $i$  in range( $0, n-1$ ):

$k = \text{indeks najmanjšega elementa v podtabli}$   
 $a[i], \dots, a[n-1]$

zamenjaj  $a[i]$  in  $a[k]$

Časovna zahtevnost: koraki

$i=0 : j=1, \dots, n-1 : n-1$

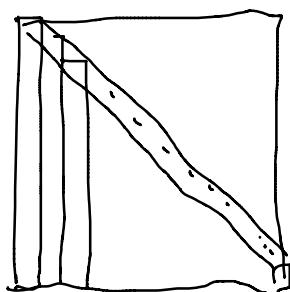
$i=1 : j=2, \dots, n-1 : n-2$

$i=2 : j=3, \dots, n-1 : n-3$

$\vdots \qquad \vdots \qquad \frac{n^2}{2} - \frac{n}{2}$

$i=n-2 : j=n-1 \dots n-1 \qquad \overbrace{1}^{1+2+3+\dots+(n-1)} = \frac{n(n-1)}{2}$

$= \Theta(n^2)$



## Urejanje z zlivanjem

Zdravje: Imamo urejeni tabeli a in b.  
Združimo ju v skupno tabelo, ki mora biti urejena.

$$a = [1, 4, 8, 10, 20, 22] \quad b = [2, 3, 7, 15, 16]$$

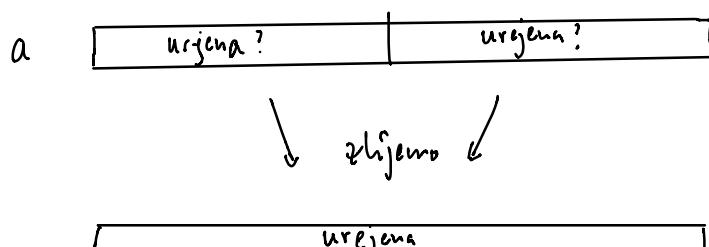
$\uparrow$     $\uparrow$     $\uparrow$   
 $i$        $j$   
 $i+j$

 $c = [1, 2, 3, 4, 7, 8]$ 

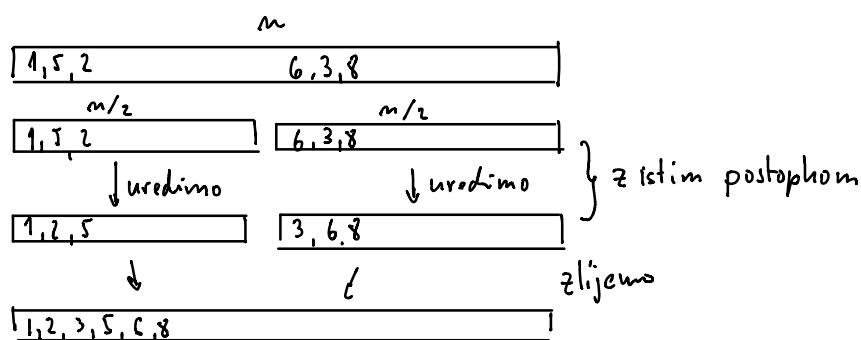
$m+m$

Casova záleženosť:  $\Theta(m+n)$

## Urejanje z zlivanjem:



## Algorithmen:

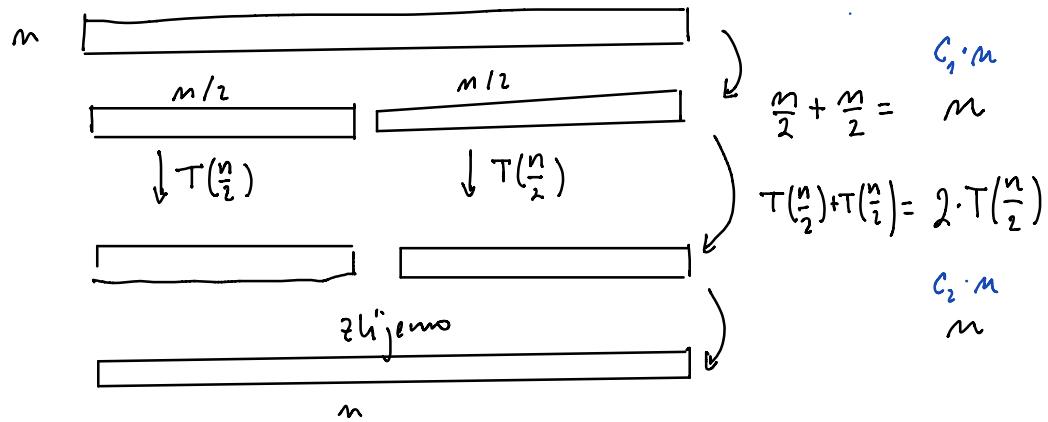


Casova zahtevost:

$T(n)$  := število korakov za urejanje z zbiranjem tabel dolžine  $n$

Krijgvordstvo

koraki



$$T(m) = m + 2 \cdot T\left(\frac{m}{2}\right) + m = 2 \cdot T\left(\frac{m}{2}\right) + 2m$$

$$(C_1 + C_2) \cdot m$$

$$2 \cdot T\left(\frac{m}{2}\right) + C \cdot m$$

$$C = 1$$

Dokimo:

$$T(m) = 2 \cdot T\left(\frac{m}{2}\right) + m \quad , \quad T(0) = 1$$

$$T(1) = 1$$

$$T(m) = 2 \cdot \underbrace{\left( 2 \cdot T\left(\frac{m}{4}\right) + \frac{m}{2} \right)}_{T\left(\frac{m}{2}\right)} + m = 4 \cdot T\left(\frac{m}{4}\right) + 2m$$

$$= 4 \cdot \left( 2 \cdot T\left(\frac{m}{8}\right) + \frac{m}{4} \right) + 2m = 8 \cdot T\left(\frac{m}{8}\right) + 3m$$

$$\vdots$$

$$= 2^3 \cdot T\left(\frac{m}{2^3}\right) + 3m$$

$$\begin{aligned}
 &= \dots + 2^k \cdot T\left(\frac{m}{2^k}\right) + k \cdot m \\
 &= 2^{\log_2 m} \cdot T(1) + (\log_2 n) \cdot m && \text{ustavite se pri } T(1) \\
 &= m \cdot T(1) + m \cdot \log_2 m && \Rightarrow k = \log_2 n \quad \begin{array}{l} \frac{m}{2^k} = 1 \\ m = 2^k \\ k = \log_2 m \end{array} \\
 &= O(m \cdot \log_2 m)
 \end{aligned}$$

Urejanje z zbiranjem je v vselj primih  $O(m \cdot \log_2 n)$

$$O(\log_a n) = O(\log_2 n) \quad \text{ker}$$

$$\log_a n = \frac{\log_2 n}{\log_2 a}$$