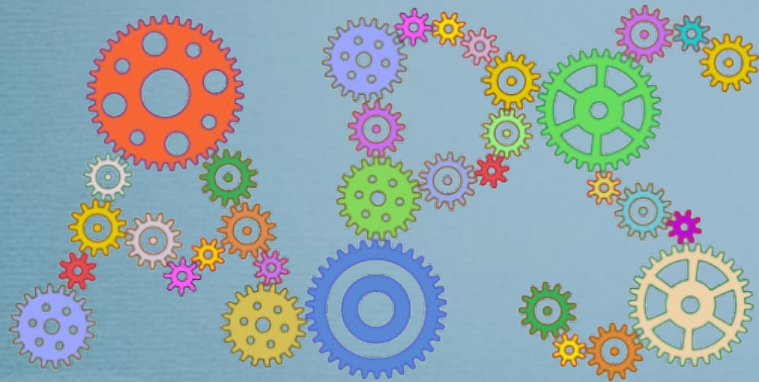


Algoritmi in podatkovne strukture 1

Visokošolski strokovni študij Računalništvo in informatika

Navadni algoritmi
urejanja zaporedja



Urejanje zaporedja

- Naloga

- **zaporedje** elementov $a = [a_0, a_1, \dots, a_{n-1}]$

3 1 4 1 5 9 2 6 5 3 5

- Rešitev

- **permutacija** $a' = [a'_0, a'_1, \dots, a'_{n-1}]$

zaporedja a

- kjer velja **urejenost elementov**

$$a'_0 \leq a'_1 \leq \dots \leq a'_{n-1}$$

sort

1 1 2 3 3 4 5 5 5 6 9

Urejanje zaporedja

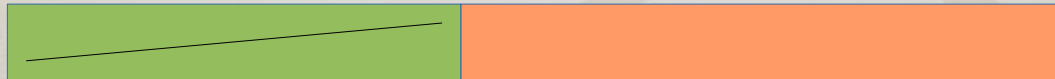
- **Nestrukturirani podatki**
 - števila (enaka dolžina)
 - nizi (različna dolžina)
 - primerjava elementa
- **Strukturirani podatki**
 - zapisi (*record, structure, class, object*)
 - ključ in (satelitski) podatek
 - primerjava ključa

3 1 4 1 5 9 2 6 5 3 5

Janez	Ana	Fran	Živa	Lana
35	24	15	66	42

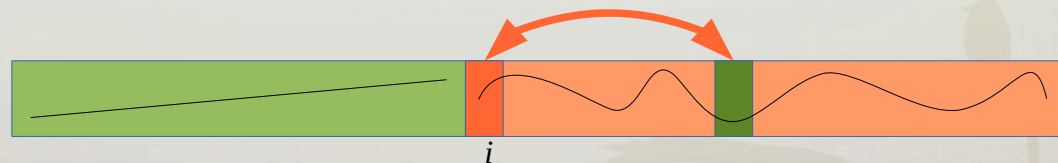
Navadna urejanja

- Navadna urejanja
 - urejanje z **izbiranjem** (*selection sort*)
 - urejanje z **vstavljanjem** (*insertion sort*)
 - urejanje z **zamenjavami** (*bubble sort*)
 - ...
 - shema navadnih urejanj
 - urejen seznam gradimo postopoma
 - levi del polja ... urejeni del seznama
 - desni del polja ... še neurejeni del seznama



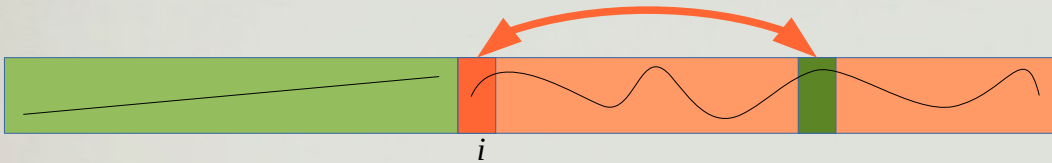
Navadno izbiranje

- Ideja algoritma
 - na vsakem koraku poiščemo najmanjši element v neurejenem delu
 - in ga *dodamo* na konec urejenega dela
 - polje: dodajanje → zamenjava
- Sled algoritma



Navadno izbiranje

- Pravilnost algoritma
 - zančna invarianta
 - vsi elementi v urejenem delu so urejeni in manjši od elementov v neurejenem delu



Navadno izbiranje

- Zahtevnost
 - št. primerjav: $n(n - 1) / 2 = \Theta(n^2)$
 - št. zamenjav: $n - 1 = \Theta(n)$
- Izboljšave?
 - hkratno iskanje min in max
 - urejanje s kopicco

Navadno izbiranje

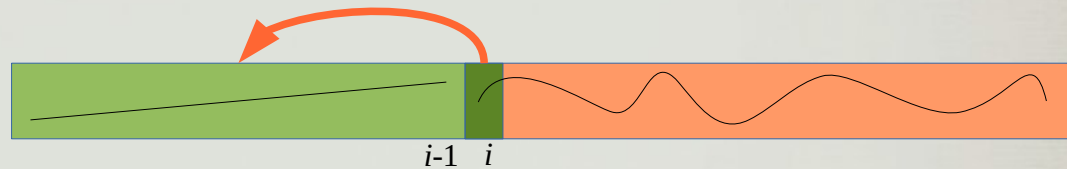
- Pseudokoda

Navadno izbiranje

```
fun selectionSort(a) is
  for i = 0 to n - 2 do
    m = i
    for j = i + 1 to n - 1 do
      if a[j] < a[m] then m = j
    swap(a, i, m)
  endfor
end
```


Navadno vstavljanje

- Ideja algoritma
 - vzamemo prvi element iz neurejenega dela in
 - ga *vstavimo* na pravo mesto v urejeni del
- Sled



Navadno vstavljanje

- Zahtevnost – št. primerjav
 - best: $n - 1 = O(n)$
 - **worst:** $n(n - 1) / 2 = O(n^2)$
 - avg: $n(n - 1) / 4 = O(n^2)$
- Izboljšave?
 - dvojiško iskanje mesta vstavljanja
 - Shellsort

Navadno vstavljanje

- Pravilnost algoritma
 - zančna invarianta (zunanja zanka)
 - v i -ti iteraciji je tabela $a[0, 1, \dots, i-1]$ urejena
 - torej $a_0 < a_1 < \dots < a_{i-1}$

Navadno vstavljanje

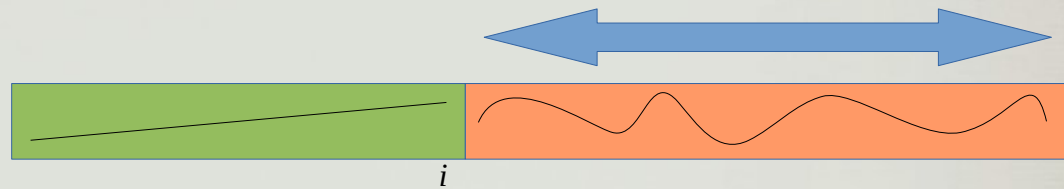
- Pseudokoda

Navadno vstavljanje

```
fun insertionSort(a) is
  for i = 1 to n-1 do
    k = a[i]
    j = i
    while j > 0 and a[j-1] > k do
      a[j] = a[j-1]
      j = j - 1
    endwhile
    a[j] = k
  endfor
end
```

Navadne zamenjave

- Ideja algoritma
 - sistematično primerjamo vsak element z vsakim
 - primerjamo paroma zaporedne elemente
- Sled



Navadne zamenjave

- Zahtevnost
 - št. primerjav: $n(n - 1) / 2 = \Theta(n^2)$
 - št. zamenjav: od 0 do $n(n - 1) / 2 = O(n^2)$
- Izboljšave?
 - detekcija urejenosti
 - izmenično urejanje

Navadne zamenjave

- Psevdokoda

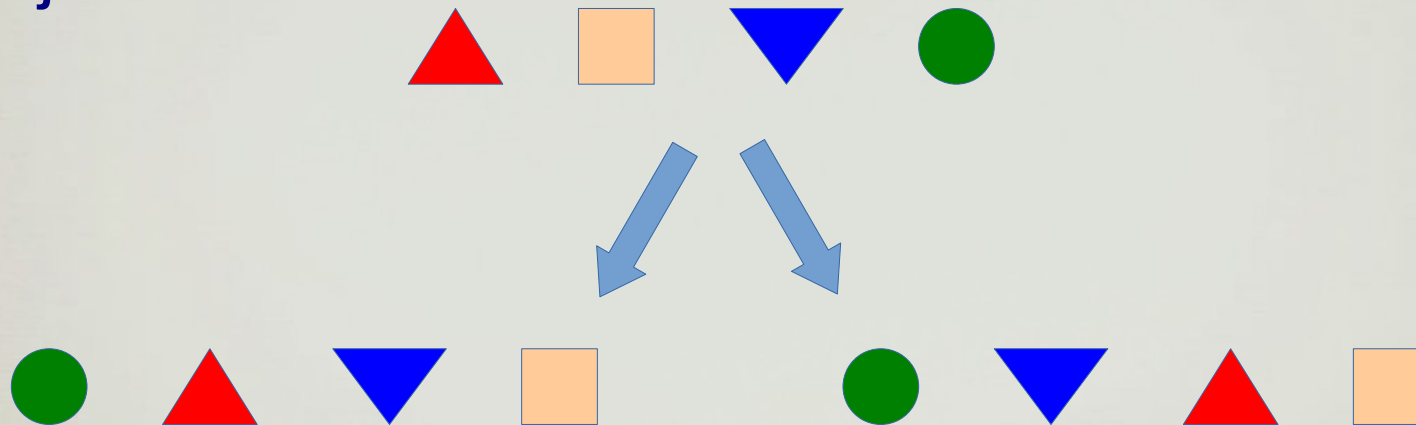
Navadne zamenjave

```
fun bubbleSort(a) is  
  for i = 1 to n - 1 do  
    for j = n - 1 to i do  
      if a[j-1] > a[j] then  
        swap(a, j-1, j)
```

Stabilnost urejanja

- Stabilnost

- strukturirani podatki (urejanje po ključu)
- ohranja prvotni vrstni red pri elementih z enakim ključem



- Uporaba

- urejanje po več ključih

Povzetek

Vrsta urejanja	Zahtevnost	Razno
Navadno vstavljanje	$O(n^2)$, best: $O(n)$	stabilno
Navadno izbiranje	$\Theta(n^2)$	
Navadne zamenjave	$\Theta(n^2)$	stabilno
Urejanje s kopico	$\Theta(n \log n)$	
Urejanje z zlivanjem	$\Theta(n \log n)$	stabilno, ni <i>in-place</i> , dodatni prostor
Hitro urejanje	$O(n^2)$, avg: $\Theta(n \log n)$	randomizacija, dodatni prostor
Urejanje s štetjem	$O(n + m)$	stabilno, končna množica
Korensko urejanje	$O(d(n + m))$	stabilno, končna množica
Urejanje s koši	$O(n^2)$, avg: $\Theta(n)$	stabilno?, enakomerno