

KĀRTOŠANA UN MEKLĒŠANA II

41.nodarbība

KĀPĒC INFORMĀCIJA JĀKĀRTO?

Ja dati ir sakārtoti, varam tajos ātrāk atrast nepieciešamo informāciju.

Kontakti tālrunī:

Anna 26262626

Apse Lauris 29292929

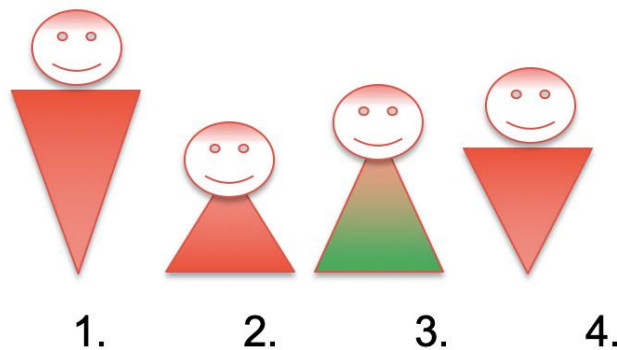
...

Kalns Ivo 61616161

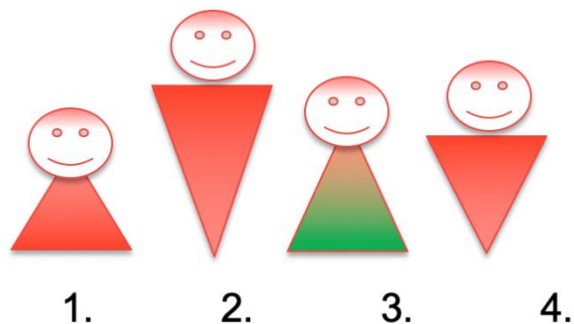
Kalsons 11111111

SELECTION SORT JEB NAIVAIS KĀRTOŠANAS ALGORITMS

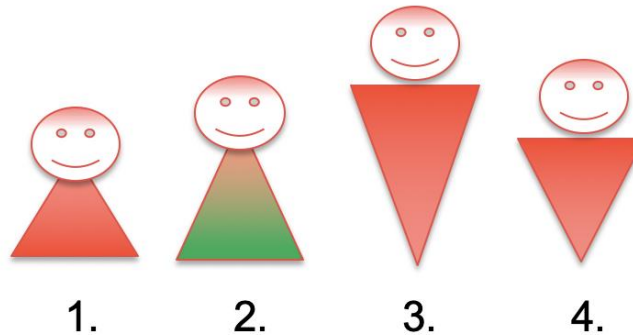
- Sakārtosim cilvēkus augošā secībā pēc to garumiem:



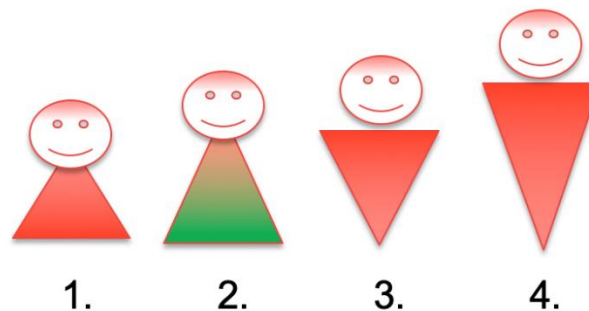
- Atrod īsāko un samaina vietām ar pirmo:



- No atlikušajiem trim atkal atrod īsāko un samaina vietām ar otro:



- No atlikušajiem diviem atkal atrod īsāko un samaina vietām ar trešo:



NAIVĀ KĀRTOŠANAS ALGORITMA NOVĒRTĒJUMS

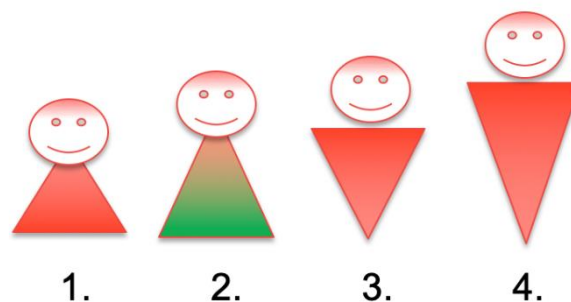
Lai sakārtotu 4 cilvēku ierindu, vajadzēja 3 reizes atrast īsāko un to mainīt vietām ar kārtējo.

n cilvēku gadījumā tas būtu jādara $n-1$ reizi.

Lai atrastu īsāko no n cilvēkiem, jāizdara $n-1$ salīdzināšana.

Taču ierindas garums, kurā jāmeklē īsākais, visu laiku samazinās.

Lai atrastu īsāko no $n-2$ cilvēkiem, jāizdara $n-3$ salīdzināšanas.



NAIVĀ KĀRTOŠANAS ALGORITMA NOVĒRTĒJUMS $\approx O(n^2)$

Kopā patērētais salīdzināšanu skaits:

1.reizē n-1

2.reizē n-2

3.reizē n-3

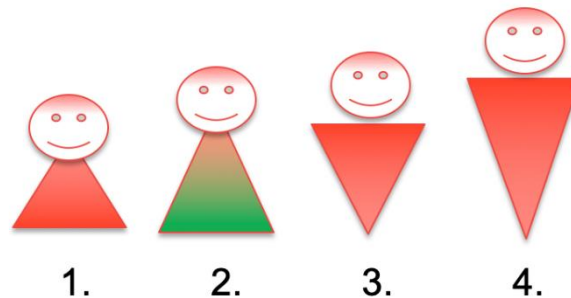
... vajag n-1 reizi!

N-1.reizē n-(n-1)=1

$$\Sigma=(n-1)+(n-2)+\dots+3+2+1=\frac{(n-1)}{2} \times (n-1+1)=$$

$$=\frac{n \times (n-1)}{2}$$

Vēl ir izdarītas arī n-1 cilvēku apmaiņa!



KĀ ATRAST MAZĀKO VĒRTĪBU TABULĀ VAI MASĪVĀ?

1. variants

min=tabula[0]

katram i no 1 līdz 6

ja tabula[i]<min tad min=tabula[i]

Rezultāts ir min vērtība!

2. variants

minVieta=0

katram i no 1 līdz 6

ja $tabula[i] < tabula[minVieta]$ tad $minVieta = i$

Rezultāts ir $tabula[minVieta]$ vērtība!

0	1	2	3	4	5	6
4	8	7	3	2	4	5

(tabula)

SELECTION SORT JEB NAIVAIS KĀRTOŠANAS ALGORITMS

katram j no 0 līdz $n-2$

$minVieta = j$

katram i no $j+1$ līdz $n-1$

ja $tabula[i] < tabula[minVieta]$ tad $minVieta = i$

Samaina vietām $tabula[minVieta]$ ar $tabula[j]$

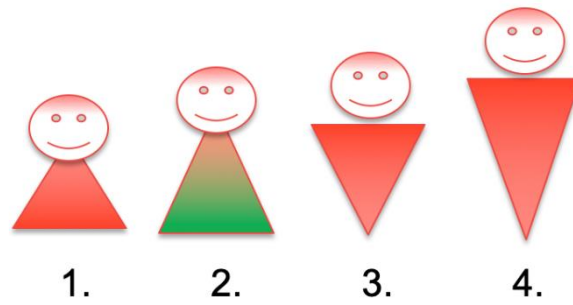
0	1	2	3	4	5	6	7		j	$minVieta$
4	8	7	3	2	4	5	1		0	7
1	8	7	3	2	4	5	4		1	4
1	2	7	3	8	4	5	4		2	3
1	2	3	7	8	4	5	4		3	5
1	2	3	4	8	7	5	4		4	7
1	2	3	4	4	7	5	8		5	6
1	2	3	4	4	5	7	8		6	6

BUBBLE SORT JEB BURBUĻA ALGORITMS – IDEJA

IR ideja!

Ja virkne ir sakārtota, katri divi blakus stāvošie arī ir sakārtoti!

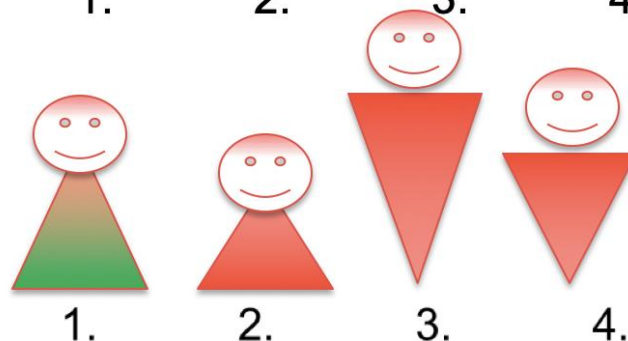
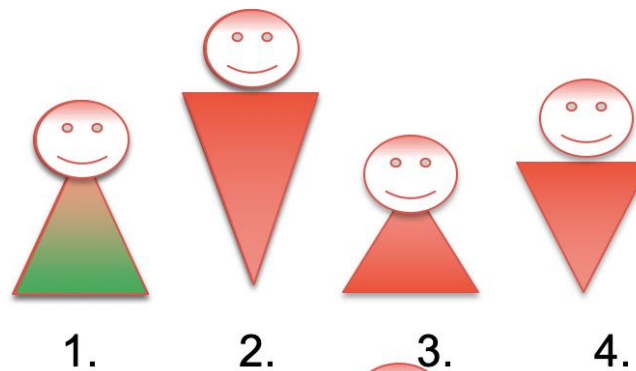
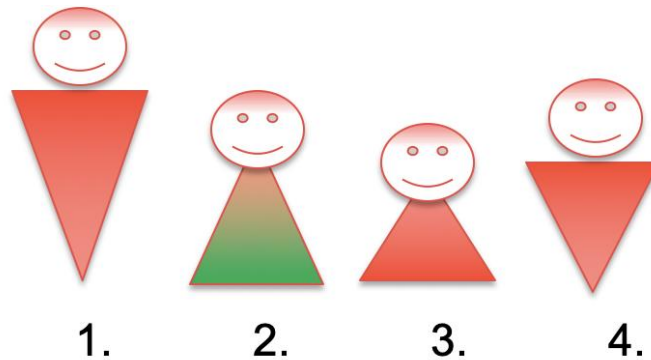
Sakārtosim blakus stāvošos, un viss būs sakārtots!



BUBBLE SORT JEB BURBUĻA ALGORITMS – PĀRU SALĪDZINĀŠANA

Salīdzinām 1. un 2. Tā kā pāris nav sakārtots, mainām 1. un 2. vietām.

Salīdzinām 2. un 3. Tā kā pāris nav sakārtots, mainām 1. un 2. vietām.

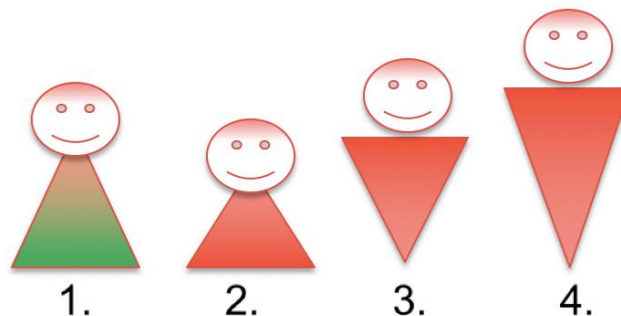
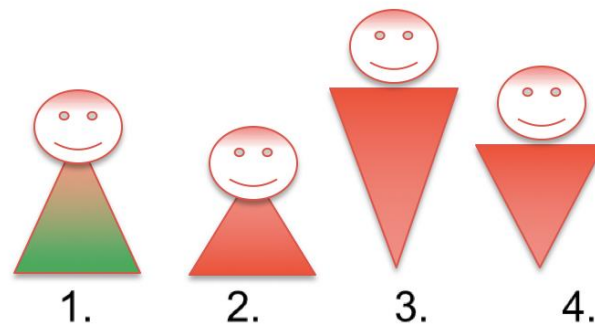


BUBBLE SORT JEB BURBUĻA ALGORITMS – GARĀKAIS VIETĀ

Salīdzinām 3. un 4. Tā kā pāris nav sakārtots, mainām 3. un 4. vietām.

Garākais ir savā vietā, taču citi vēl ne!

Vienā «caurgājienā» viens noteikti nostājas savā vietā. Ja to izdara vēl kāds, tā ir laimīga sakritība.



BUBBLE SORT JEB BURBUĻA ALGORITMS – TURPINĀM

Taisām vēl vienu «caurgājienu», bet tikai līdz trešajam, jo pēdējais jau ir savā vietā!

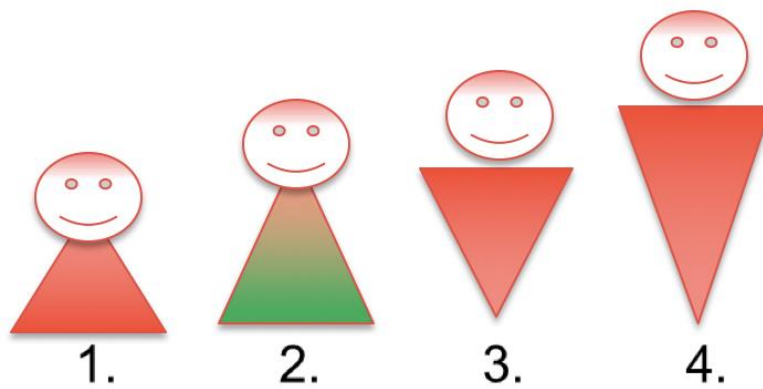
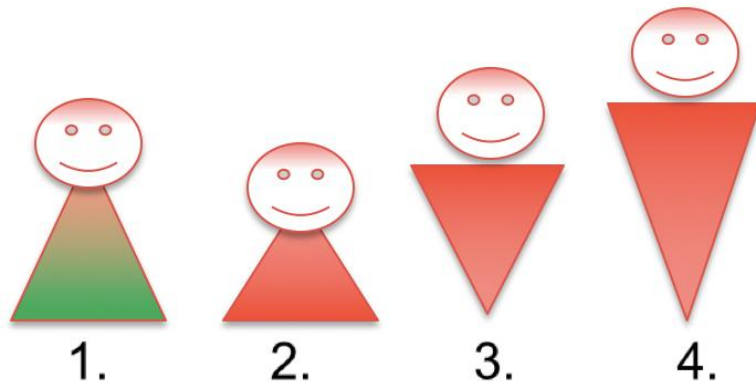
Salīdzinām 1. un 2. Tā kā pāris nav sakārtots, mainām 1. un 2. vietām.

Viss ir kārtībā, bet var arī nebūt, tāpēc turpinām.

Salīdzinām 2. un 3. Pāris ir sakārtots, neko nedarām.

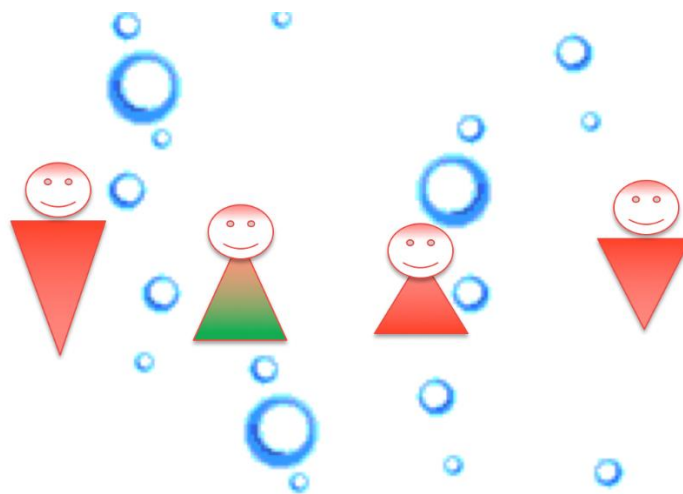
Caurgājiens beidzies. Viss ir sakārtots, bet varēja būt citādi.

Garantēti n cilvēku virkni var sakārtot ar $n-1$ caurgājieniem!



KĀPĒC BURBUĻA ALGORITMS?

Tāpēc, ka lielākais visu laiku kā tāds burbulis peld uz augšu..



BURBUĻA ALGORITMS N SKAITĻIEM

katram j no 0 līdz $n-2$

katram i no 1 līdz $n-j-1$

ja $tabula[i-1] > tabula[i]$ tad

samaina $tabula[i-1]$ ar $tabula[i]$

0	1	2	3	4	5	6
4	8	7	3	2	4	5

(tabula)

0	1	2	3	4	5	6	komentārs
4	8	7	3	2	4	5	sākums
4	7	3	2	4	5	8	1.caurgājiena rezultāts
4	3	2	4	5	7	8	2.caurgājiena rezultāts
3	2	4	4	5	7	8	3.caurgājiena rezultāts
2	3	4	4	5	7	8	4.caurgājiena rezultāts
2	3	4	4	5	7	8	5.caurgājiena rezultāts
2	3	4	4	5	7	8	6.caurgājiena rezultāts

UZLABOTS BURBUĻA ALGORITMS N SKAITĻIEM

$j=0$

atkārtot

sakārtots=true

katram i no 1 līdz $n-j-1$

ja $tabula[i-1] > tabula[i]$ tad

samaina $tabula[i-1]$ ar $tabula[i]$

sakārtots=false

$j=j+1$

kamēr nav sakārtots un $j < n-1$

0	1	2	3	4	5	6
4	8	7	3	2	4	5

(tabula)

0	1	2	3	4	5	6	komentārs
4	8	7	3	2	4	5	sākums
4	7	3	2	4	5	8	1.caurgājiena rezultāts
4	3	2	4	5	7	8	2.caurgājiena rezultāts
3	2	4	4	5	7	8	3.caurgājiena rezultāts
2	3	4	4	5	7	8	4.caurgājiena rezultāts
2	3	4	4	5	7	8	nekas nav mainīts, beidzam darbu!

BURBUĻA ALGORITMA EFEKTIVITĀTE

Algoritms	Salīdzināšanas	Maiņas
Naivais kārtošanas algoritms	$(n-1)+(n-2)+\dots+3+2+1=$ $n(n-1)/2$ $O(n^2)$	$1+1+\dots+1=$ $1(n-1)$ $O(n)$
Burbuļa algoritms	$(n-1)+(n-2)+\dots+3+2+1=$ $n(n-1)/2$ $O(n^2)$	$(n-1)+(n-2)+\dots+3+2+1=$ $n(n-1)/2$ $O(n^2)$

Sliktākajā gadījumā naivais algoritms ir labāks, jo izdara mazāk maiņas operāciju.

Taču statistiski vidēji Burbuļa algoritms nostrādā ātrāk. Burbulis ir informēts!