

# ① BUBBLE SORT SIMULATSIOON

- Võrdleb 2 arvu ning vahetab suurema paigutuse, kui

LOEND : [ 64; 34; 25; 12; 22; 11; 90 ]

1. SAMM 64 & 34  $64 > 34 \rightarrow$  VAHETAN KOHAD

loend: [ 34; 64; 25; 12; 22; 11; 90 ]

2. SAMM 64 > 25  $\rightarrow$  VAHETAN KOHAD

loend: [ 34; 25; 64; 12; 22; 11; 90 ]

3. SAMM 64 > 12

loend: [ 34; 25; 12; 64; 22; 11; 90 ]

4. SAMM 64 > 22

loend [ 34; 25; 12; 22; 64; 11; 90 ]

5. SAMM 64 > 11

loend [ 34; 25; 12; 22; 11; 64; 90 ]

6. SAMM & loend [ 25; 34; 12; 22; 11; 64; 90 ]

7. SAMM & loend [ 25; 12; 34; 22; 11; 64; 90 ]

8. SAMM & loend [ 25; 12; 22; 34; 11; 64; 90 ]

9. SAMM & loend [ 25; 12; 22; 11; 34; 64; 90 ]

10. SAMM & loend [ ~~25~~ 12; 25; 22; 11; 34; 64; 90 ]

11. SAMM & loend [ 12; 22; 25; 11; 34; 64; 90 ]

12. SAMM & loend [ 12; 22; 11; 25; 34; 64; 90 ]

( 64 < 90 )

( 34 > 25 )

( 34 > 12 )

( 34 > 22 )

( 34 < 64 )

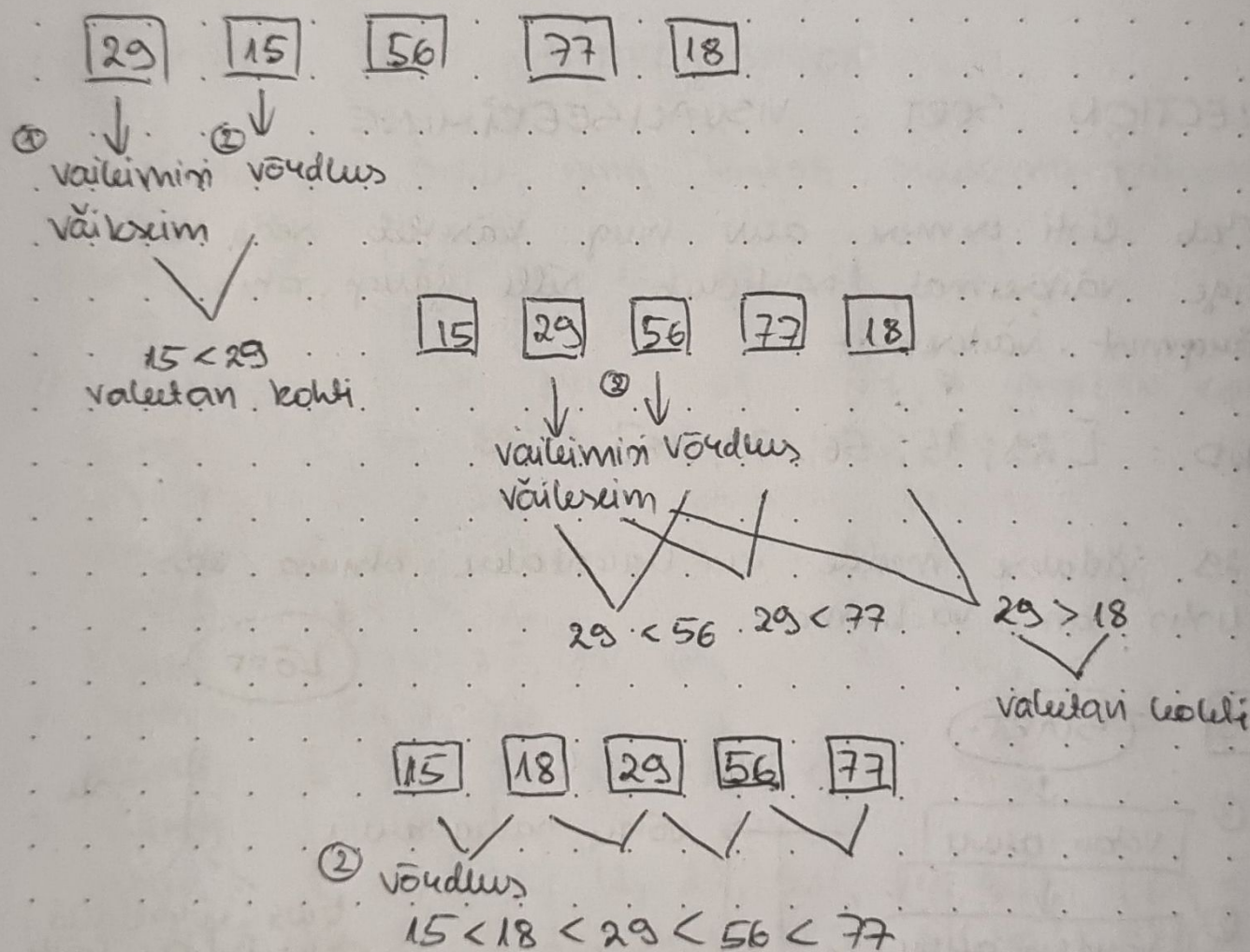
( 25 > 12 )

( 25 > 22 )

( 25 > 11 )



1. võtan arvu 29
2. võtlen arvu 29 arvuga 15
3. kas  $29 < 15$  - ei  $29 > 15$
- ~~2. võtlen arvu 29 a~~
3. võtan arvu 15 esimesele kohale
2. võtlen listis järgmist arvu, nüüd on rehtes uuesti 29 järgmise arvuga 56  $29 < 56$
2. võtlen arvu 29 77-ga  $29 < 77$
2. võtlen arvu 29 18-ga  $18 < 29$
3. võtan arvu 18 teisele kohale





13. SAMM & loend [12, 22, 11, 25, 34, 64, 90] (12 < 22)  
 14. SAMM & loend [12, 11, 22, 25, 34, 64, 90] (22 > 11)  
 15. SAMM & loend [~~12~~ 11, 22, 25, 34, 64, 90] sorteeritud

lahti on kirjutatud iga muutus, mida Bubble sort tegema peab.

1. võtlen kahe arvu
2. vahetan kohad
3. kui sama, uida sammus 1 ja loendan uii kausa leini iga paremal arv arv on suurem.

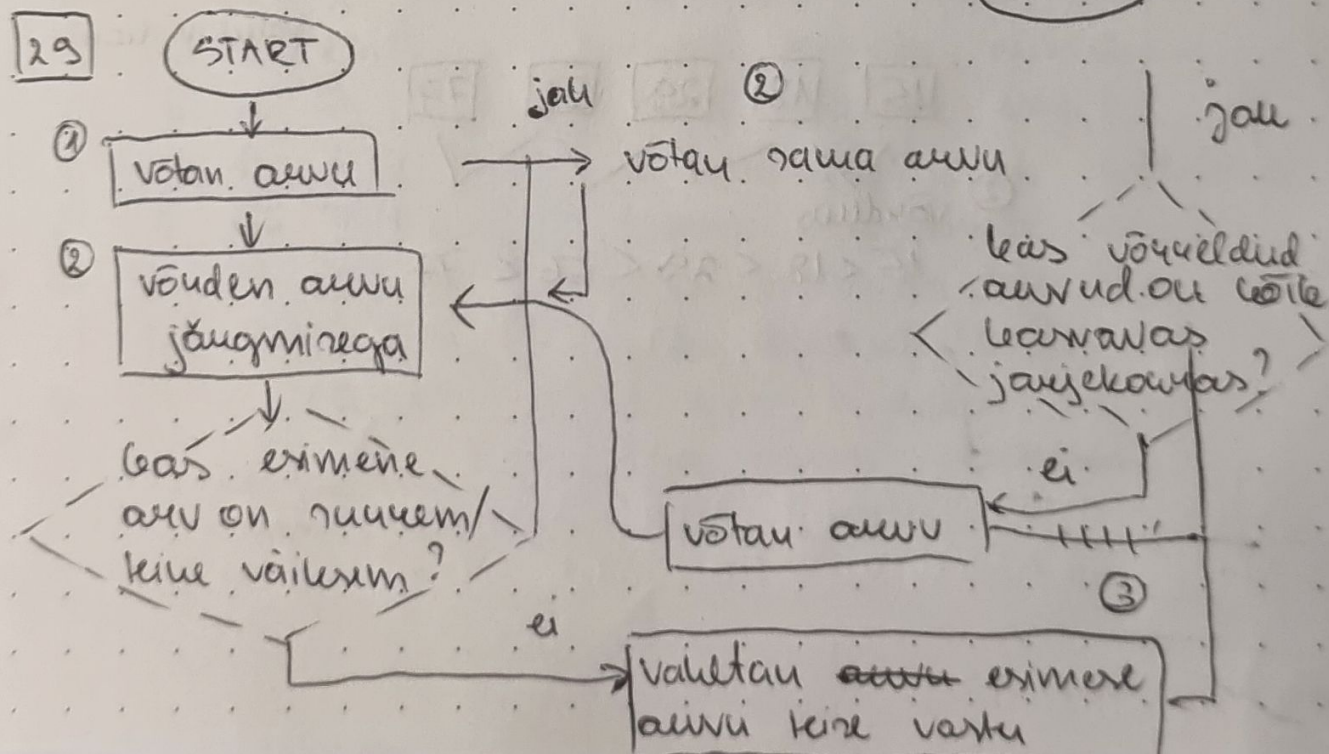
## ② SELECTION SORT VISUALISEERIMINE

- võtab listi esimese arv ning käsitseb seda kui kõige väikseimat, sorteerib selle järgi, otis järgmist väiksemat

LOEND : [29; 15; 56; 77; 18]

1. 29 jäetakse meelde ja laekatakse ohima kas listis on väiksemat

LÕPP





## ④ AJAKOMPLEKSUSTE ANALÜÜS

$n^2$

- a) Halvimal juhul ajakompleksus  $O(n^2)$  on eelmainitud koorjal - Bubble, Insertion või Selection Sortil.
- b) Kui vaatata üldiselt milline oleks parim viis Counting Sort - võimaldab juba etteantud vahemikus (1-100) sorteerida mingi viis peab loeua ainult loendis elementide erinevust. Lisaks võib see hästi just väikeste vahemikuga täisarvude jaoks.

Kui aga põhineda eelmainitud sortimisalgoritmide viis Selection Sort? ~~See~~ Tegelikult neil ei tundu suurt vahet olevat ajaline keerukus ja mahu poolest.

## ⑤ STABIILSUS JA ADAPTIIVSUS SORTIMISEL

- a) Stabiilne on ~~stabiilne~~ sortimisalgoritm, mis ~~jätaks~~ jätab samasuguse järjekorraga elementid peale sorteerimist samasse järjekorda.

NT.  $[5; 7; 11; 5; 8; 5]$  - LOEND  
 1 2 3 4 5 6 - indeksid, mis näitavad kuis element algusest listis paikneb

$[5; 5; 5; 7; 8; 11]$  - sorteeritud loend  
 1 4 6 ← indeksid

Seda saaks edukalt teha Insertion Sort sortimisalgoritmiga kasutades. (see on stabiilne)



## BOONUS

LOEND [8, 3, 5, 4, 7, 6, 2]

a) Bubble Sort [3, 5, 4, 7, 6, 2, 8]

b) Selection Sort [2, 8, 3, 5, 4, 7, 6]

c) Insertion Sort [8, 5, 3, 4, 7, 6, 2]

## ③ INSERTION SORT PRAKTIKAS

LOEND [12, 11, 13, 5, 6, 7]

– võidlen auru sellest vasakul ja liigutan parempoolse auru vasemale kuni see on väiksem

[12] [11] [13] [5] [6] [7]

~~[12] [11] [13] [5] [6] [7]~~

11 12 13 5 6 7

11 12 5 13 6 7

11 12 5 6 13 7

11 12 5 6 7 13

11 5 12 6 7 13

11 5 6 12 7 13

11 5 6 7 12 13

5 11 6 7 12 13

5 6 7 11 12 13

siin jäi üles  
samm vahule

1. kui loend on osaliselt  
sorteeritud läheb loendi  
sorteerimises suhteliselt  
vähe samme vaja

2. kergelt saab suhteliselt  
lihtsasti elementide kohad  
üha vahetada, kui need on  
juba osaliselt sorteeritud

3. pole vajadust mingit  
osa uuesti sorteerida

b) Adaptiivsus. Sorteerimisalgoritmites saab luua algoritmi mille kiirus sõltub sellest, kas loend on juba sorteeritud osaliselt või mitte. Näiteks liini loend on juba pooleldi sorteeritud väga vähe ühesandus n. 3. Insertion Sorti puhul, mis tuleb teha vähem lööd. doengus. 1 loend tabeli puhul võib esmaimist kolmest algoritmist adaptiivsus luua vaid Insertion ja Bubble Sorti, sest neil on parimad juhul ajaline keerukus  $O(n)$  ja kehtemine/vahimale  $O(n^2)$ . Samas kui Selection Sortil on kõige sama -  $O(n^2)$ .