

# Alberta koledža

## Studiju kurss "Datu struktūras un algoritmi"

vieslektors Egmonts Treiguts

### Patstāvīgo studiju 1. kontroldarbs

Iepazīties ar šķirošanas algoritmiem. Izstrādāt lietojumprogrammu šķirošanas algoritma ātrdarbības analīzei. Izpētīt kādai *big O notation* atbilst algoritms, salīdzināt ar publiski pieejamiem datiem.

Darba uzdevumi:

- 1) Izvēlēties jebkuru šķirošanas algoritmu un realizēt to C++ OOP kodā.
- 2) Pārdomāt kādas objektu klases ir nepieciešamas eksperimentu veikšanai. Izveidot nepieciešamās objektu klases, izstrādāt nepieciešamās metodes.
- 3) Sākotnējie dati ir viens CSV datu fails ar gadījumu veidā šķirotiem datiem. Ir jābūt metodei tā ielādēšanai programmā.
- 4) Lietojumprogrammas datu modeļa pamatā ir trīs masīvi ar teksta rindām. Katra teksta rinda ir gadījuma simbolu virkne, garums no 10 līdz 20 simboliem.
- 5) Visiem trijiem masīviem jāsaturs vienu un to pašu virkni. Elementu skaits masīvos ne mazāk par 10000.
  - a) Pirmajiem diviem masīviem ir jābūt iepriekš sašķirotam – vienam augošā secībā, otram – dilstošā. Trešajam masīvam jāsaturs gadījuma veidā šķirotām teksta rindām.
  - b) Šķirošana programmā vienmēr notiek augošā secībā. Pirmais masīvs simbolizēs "ātrāko" gadījumu, otrais – "lēnāko", trešais – "vidējo".
- 6) Sākotnējos datus var sagatavot, piemēram, MS Excel programmā un eksportēt uz CSV formāta failu. Gadījuma teksta rindu var iegūt kombinējot funkcijas =CHAR(RANDBETWEEN(65;90)) un concat(). Gadījuma šķirošanu var iegūt pieliekot blakus datu ailei ar rand() funkciju ģenerētus skaitļus un tad sašķirotot tabulu pēc šīs ailes.
- 7) Lietojumprogrammai ir vairākas reizes jāšķiro dati no visiem trijiem masīviem, katru reizi piefiksējot šķirošanas algoritma izpildes laiku. Šim nolūkam paredzēt metodes un īpašības rezultātu glabāšanai.
- 8) Lai iegūtu šķirošanas algoritma izpildes laika  $O()$  atkarību no šķirojamā datu apjoma ( $n$ ), katru datu masīvu jāšķiro vairākas reizes (vismaz 10) mainot datu apjomu, piemēram, var izvēlēties šādus datu apjomus ( $n$ ): 10, 20, 50, 100, 200, 500, 1000, 2000, utt.
- 9) Lietojumprogrammas izpildes rezultātā (klases atbilstošā metode) jāizveido teksta fails CSV formātā ar šādiem laukiem:
  - a) datu apjoms ( $n$ )
  - b) "ātrākā" masīva šķirošanas izpildes laiks
  - c) "lēnākā" masīva šķirošanas izpildes laiks
  - d) "vidējā" masīva šķirošanas izpildes laiks
- 10) Rezultāta teksta failu ielādēt MS Excel lietojumprogrammā un izveidot diagrammu (Scatter) kurā attēlot visu trīs datu masīvu šķirošanas laikus. Lietojot MS Excel Trendline atrast aproksimācijas

funkciju (lineāra, logaritmiska, pakāpes) katrai datu sērijai. Novērtēt aproksimāciju funkciju koeficientus un salīdzināt ar izvēlētajā šķirošanas algoritma zināmo  $O()$  notāciju.

**Norādījumi:**

- Programmēšanas valoda C++.
- Lai iegūtu uzskatāmus rezultātus, rekomendējamais datu masīva apjoms ir 10000 – 20000 ieraksti.
- Kodam ir jābūt ar atbilstīgiem komentāriem.
- Projekta katalogu arhivēt (ZIP). Arhīvam pievienot sākotnējos datus (CSV fails), rezultāta failu (CSV) un MS Excel failu ar diagrammu.
- Arhīvu ielādēt e-vidē