

1. Programmēšanas darbs

Apraksts

Lai atrastu iespējami mazāko celiņu skaitu un celiņus, kuru malā esošos kokos Pūkam ievietot medus krājumus, diezgan parocīgi ir izmantot *breadth-first search* algoritmu visu taciņu (celiņu kopu) no Pūka mājiņas līdz Sivēna mājiņai atrašanai, paralēli piefiksējot, cik reižu pa katru celiņu bijis jāiet, un beigās no katras taciņas celiņiem izvēlēties vienu celiņu, pa kuru kopumā ir bijis jāiet visvairāk.

Programmas kods rakstīts valodā JAVA. Lai palaistu Windows videi paredzēto programmu, kā parametrs jāpadod apstrādājamais teksta fails, piemēram,

- puks.exe src/ex1.txt
- puks.exe src/ex2.txt
- puks.exe src/ex3.txt

Algoritma soļi:

1. Paņem Pūka mājiņai tuvākos krustojumus
 - 1.1. Katram krustojumam pārbauda, vai tas gadījumā nav Sivēna mājiņa
 - 1.1.1. Ja ir Sivēna mājiņa
 - 1.1.1.1. Pievieno to aktuālajai taciņai
 - 1.1.1.2. Saglabā aktuālo taciņu visu taciņu sarakstā
 - 1.1.1.2.1. Katru taciņas celiņu vai nu pievieno celiņu sarakstam ar vērtību 1, vai, ja celiņš jau ir sarakstā, palielina tā vērtību par 1
 - 1.1.2. Ja nav Sivēna mājiņa
 - 1.1.2.1. Pievieno to aktuālajai taciņai
 - 1.1.2.2. Izsauc 1. Ar aktuālo krustojumu Pūka mājiņas vietā
2. Apstaigā katru taciņu
 - 2.1. Apstaigā taciņas celiņus
 - 2.1.1. Katram celiņam pievieno vērtību
 - 2.2. Rezultātu sarakstam pievieno celiņu ar vislielāko vērtību (ja tā vēl nav sarakstā)
3. Izdrukā rezultātu saraksta izmēru un katru tajā esošo celiņu

Sarežģītības novērtējums

Tā kā *breadth-first search* algoritms ir novērtējams ar $O(E+V)$ un šis algoritms, papildus *breadth-first search* algoritmam, vēl apstaigā visus celiņus jeb grafa šķautnes, tad šis algoritms novērtējams ar $O(2E+V)$.