

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
INFORMATIKOS KATEDRA

Kursinis projektas

Rikiavimo tobulinimas genetiniais algoritmais
(Improving sorting with genetic algorithms)

Atliko: 4 kurso 2 grupės studentas

Deividas Zaleskis (parašas)

Darbo vadovas:

Irmantas Radavičius (parašas)

Vilnius
2021

Turinys

Ivadas	2
1. Šelo algoritmas ir jo variantai	3
1.1. Šelo algoritmas	3
1.2. Šelo algoritmo variantai	3
Išvados	4

Įvadas

Čia bus įvadas (kai prisiversiu parašyt).

Darbo **tikslas**: pritaikyti genetinius algoritmus rikiavimo algoritmų generavimui.

Darbo uždaviniai:

- Atlikti Šelo algoritmo ir jo variantų literatūros analizę.
- Atlikti genetinių algoritmų literatūros analizę.
- Nustatyti kriterijus rikiavimo algoritmų efektyvumui įvertinti.
- Paruošti aplinką rikiavimo algoritmų generavimui.
- Pasitelkiant genetinius algoritmus sugeneruoti rikiavimo algoritmus.
- Paruošti aplinką rikiavimo algoritmų tarpusavio palyginimui.
- Atliekant eksperimentus įvertinti sugeneruotų ir klasikinių rikiavimo algoritmų efektyvumą.

Šis darbas sudarytas iš 7 skyrių. Pirmame skyriuje atliekama Šelo algoritmo ir jo variantų literatūros analizė. Antrame skyriuje atliekama genetinių algoritmų literatūros analizė. Trečiame skyriuje nustatomi kriterijai rikiavimo algoritmų efektyvumui įvertinti. Ketvirtame skyriuje paruošiama aplinka rikiavimo algoritmų generavimui. Penktame skyriuje pasitelkiant genetinius algoritmus generuojami rikiavimo algoritmai. Šeštame skyriuje paruošiama aplinka rikiavimo algoritmų tarpusavio palyginimui. Septintame skyriuje atliekant eksperimentus įvertinamas sugeneruotų ir klasikinių rikiavimo algoritmų efektyvumas.

1. Šelo algoritmas ir jo variantai

1.1. Šelo algoritmas

Algorithm 1 Vadovėlinis Šelo rikiavimo algoritmas

```

1: foreach  $gap$  in  $H$  do
2:   for  $i \leftarrow gap + 1$  to  $N$  do
3:      $j \leftarrow i$ 
4:      $temp \leftarrow S[i]$ 
5:     while  $j > gap$  and  $S[j - gap] > S[j]$  do
6:        $S[j] \leftarrow S[j - gap]$ 
7:        $j \leftarrow j - gap$ 
8:     end while
9:      $S[j] \leftarrow temp$ 
10:   end for
11: end for

```

1.2. Šelo algoritmo variantai

Algorithm 2 Patobulintas Šelo rikiavimo algoritmas

```

1: foreach  $gap$  in  $H$  do
2:   for  $i \leftarrow gap + 1$  to  $N$  do
3:     if  $S[i - gap] > S[i]$  then
4:        $j \leftarrow i$ 
5:        $temp \leftarrow S[i]$ 
6:       repeat
7:          $S[j] \leftarrow S[j - gap]$ 
8:          $j \leftarrow j - gap$ 
9:       until  $j \leq gap$  or  $S[j - gap] \leq S[j]$ 
10:       $S[j] \leftarrow temp$ 
11:     end if
12:   end for
13: end for

```

Išvados

Išvadose visą darbą sukišam į porą puslapių, tad čia gana svarbi dalis.