VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
INFORMATIKOS KATEDRA

Kursinis darbas

Rikiavimo tobulinimas genetiniais algoritmais

(Improving sorting with genetic algorithms)

Atliko: 3 kurso 2 grupės studentas
Deividas Zaleskis (parašas)

Darbo vadovas:
lekt. Irmantas Radavičius (parašas)

Vilnius
2021

Turinys

Įvadas

Darbo tikslas: **pritaikyti genetinius algoritmus Šelo algoritmo tarpų sekoms generuoti**.

Darbo uždaviniai:

- Nustatyti kriterijus tarpų sekų efektyvumui įvertinti.
- Paruošti aplinką eksperimentų vykdymui.
- Naudojant genetinius algoritmus sugeneruoti pasirinktas tarpų sekas.
- Atliekant eksperimentus įvertinti sugeneruotų ir pateiktų literatūroje tarpų sekų efektyvumą.

Duomenų rikiavimas yra vienas pamatinių informatikos uždavinių. Matematiškai jis formuluojamas taip: duotai baigtinei palyginamų elementų sekai $S = (s_1, s_2, ..., s_n)$ surasti tokį kėlinį, kad pradinės sekos elementai būtų išdėstyti didėjančia (mažėjančia) tvarka [RB13]. Rikiavimo uždavinys yra aktualus nuo pat kompiuterių atsiradimo ir buvo laikomas vienu pagrindinių uždavinių, kuriuos turėtų gebėti spręsti kompiuteris [Knu70]. Rikiavimo uždavinio sprendimas dažnai padeda pagrindą efektyviam kito uždavinio sprendimui, pavyzdžiui, atliekant paiešką sąraše, galima naudoti dvejetainės paieškos algoritmą tik tuo atveju, kai sąrašas jau yra išrikiuotas. Kadangi rikiavimo uždavinys yra fundamentalus, jam spręsti egzistuoja labai įvairių algoritmų.

Rikiavimo algoritmų yra įvairių rūšių: paremti palyginimu (rikiuojama remiantis elementų palyginimu, o ne bendromis žiniomis apie duomenis), stabilūs (lygių elementų tvarka nekeičiama), rikiuojantys vietoje (naudoja tik $O(1)$ papildomos atminties), etc [RB13]. Asimptotinis rikiavimo algoritmų sudėtingumas laiko atžvilgiu taip pat skiriasi: bogo rikiavimo (angl. bogosort) [GHR07] blogiausiu atveju atliekamas palyginimų skaičius yra neribotas, rikiavimas įterpimu (angl. insertion sort) blogiausiu atveju atlieka $O(n^2)$ palyginimų [BFM06], o asimptotiškai optimalus krūvos rikiavimas (angl. heapsort) [For64] atlieka $O(n\,log\,n)$ palyginimų [SS93]. Nepaisant rikiavimo algoritmų įvairovės, nėra algoritmo, kuris būtų geriausias visais atvejais, kadangi praktinis efektyvumas priklauso nuo daugelio veiksnių.

Šelo rikiavimo algoritmas (angl. shellsort) [She59] yra paremtas palyginimu, rikiuojantis vietoje ir nestabilus. Šelo algoritmą galima laikyti rikiavimo įterpimu modifikacija, kuri lygina ne gretimus, o toliau vienas nuo kito esančius elementus, taip paspartindama jų perkėlimą į galutinę poziciją. Pagrindinė algoritmo idėja - išskaidyti rikiuojamą seką S į posekius $S_1, S_2, ..., S_n$, kur kiekvienas posekis $S_i = (s_i, s_{i+h}, s_{i+2h}, ...)$ yra sekos S elementai, kurių pozicija skiriasi h. Seka yra h-išrikiuota, jei išrikiuoti visi posekiai S_i su tarpu h.

1. Pagrindinė tiriamoji dalis

Pagrindinėje tiriamojoje dalyje aptariama ir pagrindžiama tyrimo metodika; pagal atitinkamas darbo dalis, nuosekliai, panaudojant lyginamosios analizės, klasifikacijos, sisteminimo metodus bei apibendrinimus, dėstoma sukaupta ir išanalizuota medžiaga.

1.1. Poskyris

Citavimo pavyzdžiai nebereikalingi.

1.1.1. Skirsnis

1.1.1.1. Straipsnis

1.1.2. Skirsnis

2. Skyrius

2.1. Poskyris

2.2. Poskyris

Išvados

Išvadose ir pasiūlymuose, nekartojant atskirų dalių apibendrinimų, suformuluojamos svarbiausios darbo išvados, rekomendacijos bei pasiūlymai.

Literatūra

[BFM06] Michael A Bender, Martin Farach-Colton ir Miguel A Mosteiro. Insertion sort is O (n log n). *Theory of Computing Systems*, 39(3):391–397, 2006.

[For64] G. E. Forsythe. Algorithms. *Commun. ACM*, 7(6):347–349, 1964-06. ISSN: 0001-0782. DOI: 10.1145/512274.512284. URL: https://doi.org/10.1145/512274.512284.

[GHR07] Hermann Gruber, Markus Holzer ir Oliver Ruepp. Sorting the Slow Way: An Analysis of Perversely Awful Randomized Sorting Algorithms. Pierluigi Crescenzi, Giuseppe Prencipe ir Geppino Pucci, redaktoriai, *Fun with Algorithms*, p.p. 183–197, Berlin, Heidelberg. Springer Berlin Heidelberg, 2007. ISBN: 978-3-540-72914-3.

[Knu70] Donald E. Knuth. Von Neumann's First Computer Program. *ACM Comput. Surv.*, 2(4):247–260, 1970-12. ISSN: 0360-0300. DOI: 10.1145/356580.356581. URL: https://doi.org/10.1145/356580.356581.

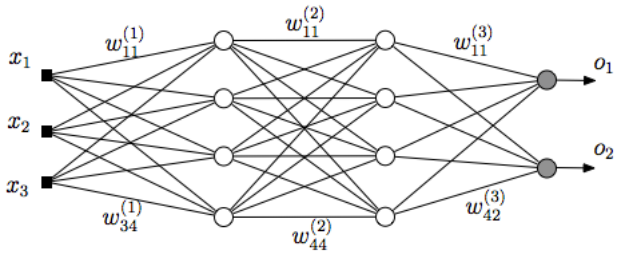
[RB13] Irmantas Radavičius ir Mykolas Baranauskas. An empirical study of the gap sequences for Shell sort. *Lietuvos matematikos rinkinys*, 54(A):61–66, 2013-12. DOI: 10.15388/LMR.A.2013.14. URL: https://www.journals.vu.lt/LMR/article/view/14899.

[She59] D. L. Shell. A High-Speed Sorting Procedure. *Commun. ACM*, 2(7):30–32, 1959-07. ISSN: 0001-0782. DOI: 10.1145/368370.368387. URL: https://doi.org/10.1145/368370.368387.

[SS93] R. Schaffer ir R. Sedgewick. The Analysis of Heapsort. *Journal of Algorithms*, 15(1):76–100, 1993. ISSN: 0196-6774. DOI: https://doi.org/10.1006/jagm.1993.1031. URL: https://www.sciencedirect.com/science/article/pii/S019667748371031X.

Priedas Nr. 1

Priedas 1



1 pav. Paveikslėlio pavyzdys