

Antroji užduotis

Tikslas: įsisavinti įvairius paieškos ir rūšiavimo algoritmus.

1. Sukurti efektyvų algoritmą ir parašyti programą, kuri rastų dviejų didelių skaičių (kelių dešimčių ar šimtų skaitmenų eilės) M ir N didžiausią bendrąjį daliklį.
2. Sukurti algoritmą ir parašyti programą, kuri dviejų matavimų tekste rastų visas vietas, kuriose yra tam tikras posekis.
3. Sukurti algoritmą ir parašyti programą, kuri surastų ilgiausią bendrąjį tolydų posekį dviejuose duotose sekose.
4. Sukurti algoritmą ir parašyti programą, kuri iš aibės S poaibių sistemos $\{s_1, s_2, \dots, s_n\}$ išrinktų minimalų skaičių poaibių, kurių sąjunga padengtų visą aibę.
5. Sukurti efektyvų algoritmą ir parašyti programą, kuri iš n duotų skaičių surastų k mažiausių.
6. Parašyti programas *heapsort*, kada naudojamos dvi skirtingos operacijos: išmesti mažiausią ir išmesti didžiausią. Palyginti jų darbą (operacijų pasiskirstymams) esant tai pačiai duomenų sekai.
7. Parašyti programą, kuri natūraliai užrašytą algebrinį reiškinį (su skliausteliais) pertvarkytų į postfiksinį reiškinį. Kurias struktūras geriau naudoti: steką, eilę, deką?
8. Parašyti programą, kuri natūraliai užrašytą algebrinį reiškinį (su skliausteliais) pertvarkytų į prefiksinį reiškinį. Kurias struktūras geriau naudoti: steką, eilę, deką?
9. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *raudonus-juodus* medžius (ne mažiau 1000 viršūnių), ir pateiktų šakų ilgių nuo šaknies iki lapų pasiskirstymą.
10. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *raudonus-juodus* medžius (ne mažiau 1000 viršūnių), ir palygintų juos su nebalansuotais dvejetainiais medžiais, pagal šakų ilgių nuo šaknies iki lapų pasiskirstymą.
11. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *raudonus-juodus* medžius (ne mažiau 1000 viršūnių), ir pateiktų viengubų ir dvigubų posūkių pasiskirstymą.
12. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *2-3-4*-medį, ir pateiktų viršūnių skaidymo sekų pasiskirstymą (skaidymo seka atsiranda, kada vienos viršūnės skaidymas iššaukia kitos viršūnės skaidymą).
13. Sudaryti efektyvų algoritmą ir parašyti programą, kuri *raudonai-juodame* medyje rastų visas viršūnes, patenkančias į tam tikrą reikšmių diapazoną.
14. Sudaryti efektyvų algoritmą ir parašyti programą, kuri *2-3-4* medyje rastų visas viršūnes, patenkančias į tam tikrą reikšmių diapazoną.
15. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų skaitmeninius paieškos medžius (ne mažiau 1000 viršūnių), ir palygintų juos su nebalansuotais dvejetainiais medžiais, pagal šakų ilgių nuo šaknies iki lapų pasiskirstymą.
16. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *daugybinius 8-skaitmeninius* medžius (8 raidžių alfabetui, ne mažiau 1000 viršūnių). Kaip tokio medžio formą įtakoja tai, kad kai kurios iš 8 raidžių pasirodo retai?
17. Sudaryti algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *daugybinius 4-skaitmeninius* medžius (ne mažiau 1000 viršūnių), ir išmestų duotą elementą.
18. Sudaryti efektyvų algoritmą ir parašyti programą, kuri naudodama atsitiktinių skaičių generatorių, sudarytų *4-trie* medžius (ne mažiau 1000 viršūnių), ir pateiktų šakų ilgių nuo šaknies iki lapų pasiskirstymą.
19. Sukurti algoritmą ir parašyti programą, kuri realizuotų *mergesort* metodą trimis su duomenimis magnetinėms juostoms, ir trimis laisvoms.

20. Sukurti algoritmą ir parašyti programą, kuri realizuotų *mergesort* metodą keturioms su duomenimis magnetinėms juostoms, ir vienai laisvai (duomenys gaunami naudojant atsitiktinių skaičių generatorių, ne mažiau 1000 skaičių).
21. Sukurti algoritmą ir parašyti programą, kuri realizuotų *išplėstinio dėstymo* metodą, jame duomenų įterpimo ir išmetimo operacijas (duomenys gaunami naudojant atsitiktinių skaičių generatorių, ne mažiau 1000 skaičių).
22. Modifikuoti *Rabin-Karp* algoritmą, kai fragmente kurioje nors vidinėje pozicijoje yra "wild card" simbolis.
23. Modifikuoti *Rabin-Karp* algoritmą, kad jo pagalba būtų galima ieškoti fragmento dvimačiame tekste.
24. Modifikuoti *Knuth-Morris-Pratt* algoritmą, kai fragmente kurioje nors vidinėje pozicijoje yra "wild card" simbolis.
25. Modifikuoti *Knuth-Morris-Pratt* algoritmą, kad jo pagalba būtų galima ieškoti fragmento dvimačiame tekste.
26. Modifikuoti *Boyer-Moore* algoritmą, kai fragmente kurioje nors vidinėje pozicijoje yra "wild card" simbolis.
27. Modifikuoti *Boyer-Moore* algoritmą, kad jo pagalba būtų galima ieškoti fragmento dvimačiame tekste.
28. Sukurti algoritmą ir parašyti programą, kuri duotoje sekoje suranda nors vieną palindromą (t.y. posėkį, kuris vienodai skaitomas iš kairės į dešinę, ir iš dešinės į kairę).
29. Modifikuoti Huffman'o algoritmą trejetainiam atvejui (kada koduojantis alfabetas susideda iš 3 raidžių).
30. Parašyti programą, kuri koduotų ir dekoduo­tu­ dvej­et­ain­į failą RLE metodu.
31. Sukurti algortimą ir parašyti programą, kuri modifikuoja RLE metodą trejetainiam failui.