

## Užduotys projektui

Tikslas: įsisavinti sudėtingesnes duomenų struktūras ir algoritmus.

1. Duotos dvi simbolių sekos. Į šias sekas tarp simbolių gali būti įterpiama simbolis \*, turintis specialią reikšmę: jei sekos simboliai yra skaitomi po vieną ir talpinami į steką, tai sutikus simbolį \*, vykdoma operacija *pop*. Reikia sukurti efektyvų algoritmą, kuris į pirmą seką įterpia simbolius \* taip, kad įvykdžius steko operacijas, yra gaunama antroji seka (arba nustatoma, kad tokie įterpimai yra neįmanomi).
2. Sukurti ADT "daugianaris", kuris skirtas operacijoms su vieno kintamojo daugianariais: dviejų daugianarių sudėtis, atimtis, daugyba, dalyba su liekana, integralas, superpozicija.
3. Sukurti efektyvų algoritmą, kuris nustato ar du duoti dvejetainiai medžiai yra izomorfiniai, t.y. ar du duoti dvejetainiai medžiai atitinka vieną ir tą patį nesutvarkytą dvejetainį medį (ar galima vieno medžio viršūnes pernumeruoti taip, kad būtų gautas kitas medis).
4. Greito rūšiavimo algoritmas (quicksort) rūšiuoja  $N$  atsitiktinių skaičių, kaip žinia kiekviename žingsnyje skaidydamas turimą skaičių seką į du posekius. Reikia rasti vidutinį skaičių posekių, kurių ilgis yra 0, 1, 2, 3, arba duotas skaičius  $k$ .
5. Intervalinė paieška (interval search) gali būti naudojama atveju, kai reikia rasti visus duotos aibės elementus, esančius nurodytame intervale. Šią problemą galima spręsti konstruojant intervalinius medžius, t.y. dvejetainius paieškos medžius su dviem rodyklėmis - į mažiausią viršūnę, ir į didžiausią viršūnę (arba su rodyklėmis į atitinkamus pomedžius). Grid method siūlo kitokį algoritmą. Reikia šį algoritmą įsisavinti, implementuoti, analizuoti. Intervalinė paieška labai dažnai sutinkama techninių brėžinių duomenų bazėse.
6. Iškiliojo kevalo (convex hull) algoritmas: plokštumoje turime  $n$  taškų aibę. Iškilusis apvalkalas yra mažiausias galimas iškilus daugiakampis, kai visi aibės taškai yra jo viduje arba ant briaunų bei viršūnių. Reikia įsisavinti Graham's scan algoritmą, jį analizuoti, implementuoti, nustatyti jo efektyvumą vidurkio atveju ir blogiausiu atveju. Iškilusis kevalas dažnai naudojamas duomenų bazių indeksams vertinti, algoritmo aprašymą galima rasti Internete.
7. Iškiliojo kevalo (convex hull) algoritmas: plokštumoje turime  $n$  taškų aibę. Iškilusis apvalkalas yra mažiausias galimas iškilus daugiakampis, kai visi aibės taškai yra jo viduje arba ant briaunų bei viršūnių. Reikia įsisavinti Jarvis march algoritmą, jį analizuoti, implementuoti, nustatyti jo efektyvumą vidurkio atveju ir blogiausiu atveju. Iškilusis kevalas dažnai naudojamas duomenų bazių indeksams vertinti, algoritmo aprašymą galima rasti Internete.
8. Iškiliojo kevalo (convex hull) algoritmas: plokštumoje turime  $n$  taškų aibę. Iškilusis apvalkalas yra mažiausias galimas iškilus daugiakampis, kai visi aibės taškai yra jo viduje arba ant briaunų bei viršūnių. Reikia įsisavinti divide-and-conquer metodą, jį analizuoti, implementuoti, nustatyti jo efektyvumą vidurkio atveju ir blogiausiu atveju. Iškilusis kevalas dažnai naudojamas duomenų bazių indeksams vertinti, algoritmo aprašymą galima rasti Internete.
9. Iškiliojo kevalo (convex hull) algoritmas: plokštumoje turime  $n$  taškų aibę. Iškilusis apvalkalas yra mažiausias galimas iškilus daugiakampis, kai visi aibės taškai yra jo viduje arba ant briaunų bei viršūnių. Reikia įsisavinti Incremental metodo algoritmą, jį analizuoti, implementuoti, nustatyti jo efektyvumą vidurkio atveju ir blogiausiu atveju. Iškilusis kevalas dažnai naudojamas duomenų bazių indeksams vertinti, algoritmo aprašymą galima rasti Internete.
10. Grid metodo pagrindu galima konstruoti dvimatę intervalinę paiešką (interval search), t.y. rasti visus duotos aibės plokštumoje taškus, esančius nurodytoje srityje. Reikia šį algoritmą įsisavinti, implementuoti, analizuoti. Intervalinė paieška labai dažnai sutinkama techninių brėžinių duomenų bazėse.
11. Dvimačiai intervalinės paieškos medžiai (two-dimensional range tree). Šie medžiai yra dinaminė adaptivi struktūra, skaidanti erdvę (plokštumą) į dvi dalis, pakaitomis lygiagrečiai  $x$ - ir  $y$ -ašiai. Reikia konstruoti dvimačius intervalinės paieškos medžius, kai plokštumą skaidoma tiesėmis, adaptuotomis aibės formai.

12. Susikertančių atkarpų paieškos (peržiūros) algoritmas. Plokštumoje duota aibė atkarpų, kai kurios iš jų gali kirstis. Konstruojant x-tree ir y-tree, sukurti algoritmą, randantį visas susikertančias atkarpas.
13. Dviejų artimiausių taškų plokštumoje radimo algoritmas (Voronojaus diagramos ir Delauney trianguliacija). Reikia sukurti, implementuoti ir analizuoti algoritmą, randantį aibės du artimiausius taškus plokštumoje, pasinaudojant minėtomis diagramomis, arba joms dualia Delauney triangulation.
14. Huffmano algoritmas žodžių fragmentams (nustatyti labiausiai tinkamus fragmentus). Kai duotas koks nors tekstas, galima manyti, kad jo alfabetą sudaro ne pavienės raidės, o dvigarsiai, trigarsiai ir pan. reikia sukurti algoritmą, kuris nustatytų labiausiai tinkamus fragmentus, taip kad Huffmano kodas būtų trumpiausias.
15. Ištirti dvejetainius paieškos medžius: įterpti įrašus į medį jų dažnumų mažėjimo tvarka, kai iš anksto žinomi paieškos raktų dažnumai; modifikuoti dvejetainius medžius, kad jie galėtų turėti pasikartojančius raktus; sudaryti rekursyvias programas, spausdinančias medžių viršūnes duota sutvarkymo tvarka.
16. Sukurti keturių spalvų medžių teoriją (kokio tipo paieškos struktūroms gali atstovauti medžiai, kai raudoname-juodame medyje spalvai žymėti yra naudojami du arba trys bitai).
17. Suffix-tree algoritmai (*Patricia* algoritmo apibendrinimas), ištirti jų savybes. Suffix-tree ir multiway suffix tree vaidina labai didelę rolę maršrutizatoriuose Internete. Jų aprašymų gausu Internete. Reikia įsisavinti tokius algoritmus, implementuoti, analizuoti jų efektyvumą (priklausomai nuo duomenų charakterio).
18. Union-Find struktūros. Reikia hierarchinėmis struktūromis (heap, medžiais ir pan.) dėstyti kompiuterio atmintyje aibes, kurių sankirta paporiui visada yra tuščia.