

Olimpiadinių algoritmavimo uždavinių analizė moksleivių ir ekspertų požiūriu

Jonas BLONSKIS (KTU), Valentina DAGIENĖ (MII), Kostas PLUKAS (KTU)

el. paštas: jonasb@soften.ktu.lt, dagiene@ktl.mii.lt, kostas.plukas@ktu.lt

Ivadas

Lietuvos nacionalinių baigiamojo etapo, Baltijos šalių ir pasaulio informatikos olimpiadų uždaviniai savo sudėtingumu mažai skiriasi [1, 3, 4, 5]. Jiems spręsti būtinos mokyklinės matematikos žinios yra nepakankamos. Papildomai reikia žinoti diskrečiosios matematikos, grafų teorijos, kombinatorikos ir diskretinio optimizavimo metodus.

Rašomų programų sudėtingumas, apimtis, uždavinio sprendimo laikas priklauso nuo pasirinktų duomenų struktūrų, gebėjimo jas susieti su sprendimo algoritmais. Programavimą gerokai apsunkina didelis duomenų skaičius. Tam reikia žinoti išorinio duomenų apdorojimo būdus. Mokykloje pasirenkamasis programavimo pagrindų modulis yra tik XI–XII klasėse, todėl mokytojai mokinius olimpiadoms turi ruošti atskirai, mokiniai turi daug dirbti savarankiškai arba mokytis kompiuterininkų mokyklose [2]. Suprantama, kad reikiamos tiek matematikos, tiek programavimo žinios privalo būti teikiamos kiek galima anksčiau. Stebint mokinius ilgą laiką, galima padaryti išvadą, kad reikalingos bazinės žinios turi būti suteiktos 5–7 klasėse. Mokytojai turi gerai išmanyti ir matematiką, ir programavimą.

1. Matematinis pasiruošimas

Šio straipsnio autoriai atliko pastarųjų trijų metų nacionalinės (N), Baltijos šalių (B) ir pasaulio (P) moksleivių informatikos olimpiadų uždavinių išsamią analizę; remiantis ja galima teigti, kad olimpiados uždaviniams spręsti reikia diskrečiosios matematikos žinių: grafų teorijos, kombinatorikos, Bulio algebros, diskretinio optimizavimo ir skaičiavimo matematikos metodų.

Iš grafų teorijos dažniausiai naudojami paieškos gilyn, paieškos platyn bei paieškos gilyn drauge naudojant grįžimą metodai. Pavyzdžiui, paieška platyn taikoma šiuose uždaviniuose: „Monetų stekai“ (B), „Sienos“ (P), „Virusai“ (N); paieška gilyn naudojant grįžimą – uždaviniuose „Bikritinis kelias“ (B), „Greičio ribojimas“ (B). Rasta nemažai uždavinių, kuriuose reikia nustatyti, ar grafas neturi kontūro („Simbolių keitimas“(N)), rasti grafo sąlyčio tašką („Pažintys“(N)), sukonstruoti grafą, kai duota jo viršūnių laipsnių seka („Teniso klubas“(N)) ir kt.

Yra daugybė kombinatorinio optimizavimo uždavinių (pvz., „Robotų judėjimas“(B), „Lenta“(B), „Skaičiai“(N) ir kt.), kuriems spręsti naudojami dinaminio programavimo

bei šakų ir rėžių metodai, o taip pat skaičiavimo tipo uždaviniai, kurių charakteringas pavyzdys gali būti „Žaidimų automatas“(B) – jis nesunkiai transformuojamas į tiesinių lygčių sistemos sprendimą.

Reikia pažymėti, kad vyresniųjų klasių moksleivių informatikos olimpiados uždaviniai reikalauja tų pačių žinių, kaip ir studentų informatikos olimpiadų uždaviniai. Paminėti metodai nenagrinėjami bendrajame matematikos kurse, todėl reikiamas žinias mokiniai turi įgyti papildomai.

2. Programavimas

Remiantis apklausos rezultatais programavimo žinias ir įgūdžius dauguma olimpiadininkų įgyja ne per pamokas. Tai gali būti būreliai, kompiuterininkų mokyklos, individualus darbas talkinant mokytojams (nebūtinai informatikos), moksleiviams bendraujant tarpusavyje. Daugeliu atvejų tos žinios nėra nuoseklios ir išsamios. Žinios apima programavimo kalbą, duomenų tipus ir struktūras, algoritmus darbui su duomenimis naudojant pasirinktas struktūras, programavimo technologijos pagrindus, veiksmus su išorinėmis duomenų struktūromis, darbą su programavimo terpe, kompiliatoriaus ypatumus. Labai svarbi žinių ir įgūdžių dalis yra matematinių sprendimo metodų realizacijos ypatumai, suderinamumas su sukurtomis duomenų struktūromis. Taigi pasiekti vidutinių rezultatų olimpiadose galima tik įsisavinus pagrindines universitetinio informatiko ruošimo žinias (1–2 kursas). To reikalauja olimpiadų rengėjai teikdami uždavinius.

Darbui su duomenimis būtina mokėti perkelti duomenis iš bylų į sukurtas duomenų struktūras, gebėti duomenis rūšiuoti, šalinti, papildyti, atlikti paiešką, filtruoti ir pan. Atskiruose masyvuose šie veiksmai nesudėtingi, tačiau susietuose loginiais ryšiais masyvų rinkiniuose tie veiksmai tampa pakankamai sudėtingais, reikalauja iš mokinių patyrimo, įgudimo, atidumo, o svarbiausia, drąsos priimti nestandartinius sprendimus rizikuojant rezultatais. Duomenų struktūros kuriamos atsižvelgiant į duotą uždavinį, todėl kuriami ir atitinkami algoritmai darbui su duomenimis. Mokiniai turi žinoti tuos pagrindinius algoritmus ir mokėti juos modifikuoti, pritaikyti konkrečioms situacijoms.

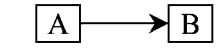
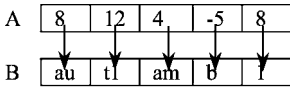
3. Duomenų struktūros

Mėgstamiausia mokinių duomenų struktūra yra masyvas. Tik nedidelei daliai uždavinių duomenims aprašyti ir laikyti pakanka kelių vienmačių arba dvimačių masyvų. Daugumai uždavinių reikia kur kas daugiau ir sudėtingesnių duomenų struktūrų.

Apdorojant duomenis būtina ne tik juos saugoti keliuose tarpusavyje susietuose masyvuose, bet ir sprendimo eigoje organizuoti dinamiškai kintančius ryšius, įrašinėti tarpinius rezultatus. Nemokėdami susieti skirtinguose masyvuose laikomus duomenis, mokiniai kuria papildomus, perteklinius masyvus. Programos tampa neracionalios, sudėtingos, jose sunku įvertinti visus užduotyse numatytus reikalavimus.

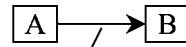
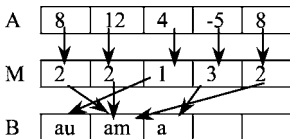
Pateikiame keletą masyvų organizavimo būdų.

Vienareikšmis atitikimas. Tai situacija, kai kiekvienai reikšmei a_i iš masyvo $A(n)$ atitinka reikšmė b_i iš masyvo $B(n)$. Tai tiesioginis atitikimas pagal indeksą. Gali būti daugiau masyvų. Juose turi būti vienodas skaičius reikšmių ir jos tarpusavyje susiejamos tuo pačiu indeksu.



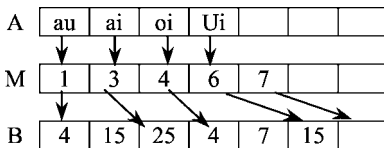
Grafinis vaizdavimas

Vienareikšmis indeksinis atitikimas. Tai situacija, kai kiekvienai reikšmei a_i iš masyvo $A(n)$ atitinka reikšmė b_j iš masyvo $B(m)$. Tai tiesioginis atitikimas pagal indeksą, tačiau masyvuose gali būti skirtingas duomenų skaičius. Duomenims susieti reikalingas papildomas indeksų masyvas M .



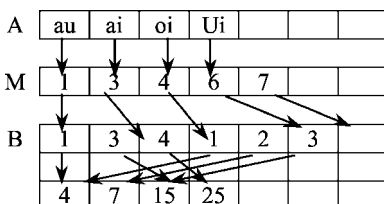
Grafinis vaizdavimas

Daugiareikšmis atitikimas. Tai situacija, kai kiekvienai reikšmei a_i iš masyvo $A(n)$ atitinka grupė masyvo $B(m)$ reikšmių. Duomenims susieti reikalingas papildomas indeksų masyvas M , kuriame surašomi kiekvienos grupės masyve B pradžios indeksai. Gretimų indeksų skirtumas nusako grupės dydį.



Grafinis vaizdavimas

Daugiareikšmis indeksinis atitikimas. Tai situacija, kai kiekvienai reikšmei a_i iš masyvo $A(n)$ atitinka grupė masyvo $B(m)$ reikšmių. Masyvo B reikšmės gali priklausyti kelioms grupėms. Duomenims susieti reikalingi du papildomi indeksų masyvai $M1$ ir $M2$. Masyve $M1$ surašomi kiekvienos grupės masyve $M2$ pradžios indeksai. Masyve $M2$ surašomi indeksai į masyvo B reikšmes.



Grafinis vaizdavimas

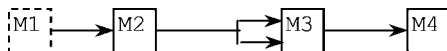
Jeigu duomenų masyvo reikšmės sutampa su indeksų reikšmėmis, tuomet tokio masyvo gali nebūti, pvz., kai masyvo A reikšmės yra 1, 2, 3, 4 ir t. t. Grafiškai vaizduojant toks masyvas pažymimas punktyrinio kontūru.

Daug uždavinių siejami su grafais. Juos vaizduoti patogiu gretutinumo arba insidencijų matricomis, programos nesudėtingos, tačiau esant dideliams duomenų skaičiams netinka, nes reikia daug kompiuterio atminties. Tuomet vartojamos masyvų sąrašinės struktūros. Pavyzdžiui, 2003 m. Nacionalinės olimpiados uždavinyje „Virusas“ [5] kompiuterių tinklas buvo vaizduojamas grafu, kurio viršūnės turėjo reikšmes T (yra virusinė apsauga) arba N (nėra apsaugos nuo virusų). Uždavinio esmė buvo nustatyti, ar virusas, startavęs nurodytoje A viršūnėje, pasieks nurodytą viršūnę B . Buvo sąlygoje pasakyta, kad jeigu virusas vienu metu pasiekia kurią nors viršūnę skirtingais keliais, tai jis dirba kaip vienas: naikina antivirusinę programą ir pats susinaikina. Virusas gali eiti ryšiu tik vieną kartą, nes jį sugadina.

Pateiktame pavyzdyje virusas, startavęs viršūnėje $A = 1$, viršūnės $B = 5$ nepasieks, nes jį blokuos 4-a viršūnė (ją vienu metu pasiekęs virusas iš 2-os ir 3-ios suveiks kaip vienas). Jeigu $A = 2$, tuomet virusas pasieks 5-ą viršūnę: pirmu žingsniu iš 2-os viršūnės atėjęs virusas sunaikins apsaugą (4-os viršūnės svoris taps N), o trečiu žingsniu (per 1-ą ir 3-ią viršūnes) atėjęs virusas nebus sulaikomas.

Olimpiadoje 23 mokiniai (iš 32) pirmuoju atveju gavo klaidingą atsakymą.

Tokio tipo grafo duomenis galima saugoti masyvuose, kurių grafinis aprašas toks:



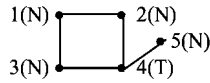
Čia $M1$ – viršūnių numerių sąrašas (numeruojamos eilės tvarka; sutampa su indekso reikšmėmis; masyvas menamas); $M2$ – viršūnių reikšmės (šiuo atveju T arba N simboliai); $M3$ – ryšių grupių pradžios indeksai masyve $M4$; $M4$ – ryšio viršūnių numeriai.

M2	N	N	N	T	N						
M3	1	3	5	7	10	11					
M4	2	3	1	4	1	4	2	3	5	4	

Jeigu briaunos turėtų svorius (realiai tai galėtų būti ryšiu tarp kompiuterių signalo sklidimo laikas), tuomet būtų reikalingas dar vienas masyvas, tiesiogiai susietas su $M4$, kuriame būtų surašytos briaunų svorio reikšmės.

Kelio tarp duotų viršūnių paieška yra vienas dažniausiai pasitaikančių veiksmų olimpiadiniuose uždaviniuose. Jeigu nėra papildomų sąlygų, tuomet geriausia taikyti bangos algoritimą – kelias randamas nesunkiai. Čia net nebūtina fiksuoti bangos sklidimo frontų. Pakanka turėti dinamiškai kintantį masyvą, iš kurio pradžios imamos viršūnės eiliniam bangos žingsniui, o gale rašomos naujai pasiektos viršūnės. Taupant atmintį darbą su masyvu galima organizuoti žiediniu principu: pasiekus pabaigą, pradėti rašyti naujas viršūnes nuo pradžios (avarinė situacija, kai naujų viršūnių sąrašas pasiekia dar nepanaudotų viršūnių sąrašą). To išvengiama ir veiksmai būna paprastesni, jei naudojamas žiedinis dinamiškas sąrašas. Tačiau retas mokinys moka dirbti su rodyklėmis. O juk $M3$ ir $M4$ pakeitus tiesinių dinamiškų sąrašų rinkiniu darbas supaprastėtų (kiekvienas i -asis

elementas iš $M2$ būtų tiesiogiai susiejamas su tiesišku dinamišku sąrašu, kuriame būtų laikomi i -os viršūnės ryšiai su kitomis).



Šiame pavyzdyje banga turi būti skleidžiama frontais. Tik eiliniam frontui pasibaigus galima keisti arba ne viršūnės reikšmę iš T į N . Reikalingi du papildomi masyvai talpinti fronto, iš kurio plinta banga, viršūnėms, ir fronto naujai pasiektų viršūnių sąrašui.

Pradžios viršūnė (1 frontas)

Antras frontas

Trečias frontas

1		
2	3	
4		

4-oji viršūnė pasiekama iš 2 ir 3, tačiau į frontą įrašoma vieną kartą. Kadangi 4-osios viršūnės reikšmę T keičiame į N , o virusas toje viršūnėje išnyksta, tai 4-oji šalinama iš fronto sąrašo. Kadangi fronto sąrašė nebeliko viršūnių, tai banga toliau nesklinda. Viršūnė $B = 5$ nepasiekta.

Išvados

Informatikos olimpiadų uždaviniams spręsti mokyklinių matematikos žinių nepakanka. Būtinai specialus mokinių ruošimas.

Mokinius olimpiadoms būtina pradėti ruošti 5–7 klasėse. Mokymo programos pagrindą turi sudaryti programavimo pagrindai ir matematiniai metodai.

Olimpiadininkai vienodai gerai turi išsivirti matematikos ir programavimo žinias, kurios maždaug atitinka universiteto 1–2 kurso informatikos programoms.

Literatūra

- [1] V. Dagienė, J. Skūpienė, Algoritmavimo uždavinių sprendimo metodų bei sudėtingumo analizė Lietuvos informatikos olimpiadose, *Liet. matem. rink.* (įteikta, rankraštis) (2003).
- [2] V. Dagienė, J. Blonskis, Programavimo mokymas išplėstiniame informatikos kurse, *Liet. matem. rink.*, **42** (spec. nr.), 229–234 (2002).
- [3] V. Dagienė, Moksleivių informatikos olimpiados: tikslai ir organizavimo principai, *Fizikos, matematikos ir informatikos mokymas reformuojamoje mokykloje*, Šiauliai, 13–17 (1997).
- [4] V. Dagienė, J. Skūpienė, *Moksleivių informatikos olimpiadų uždaviniai*, I, II d., TEV, Vilnius (1999, 2001).
- [5] Lietuvos moksleivių informatikos olimpiados, Adresas internete: <http://aldona.mii.lt/pms/olimp>

Analysis of algorithmic problems solved in Olympiads: stands of students and experts

J. Blonskis, V. Dagienė, K. Plukas

Olympiads (National, Baltic, International) in Informatics are well-known among students. Various algorithmic problems are given to students. The paper deals with the problems from the last three years. Methods of solving problems are discussed from mathematical and programming perspectives. The main attention is paid to selection, development and application of proper data structures.