

Algoritmavimo uždavinių sprendimų vertinimo problemos

Valentina DAGIENĖ, Viktoras DAGYS (MII)

el. paštas: dagiene@ktl.mii.lt, dagys@ktl.mii.lt

Reziumė. Straipsnyje analizuojami algoritmavimo uždavinių sprendimų vertinimo būdai, naudojami Lietuvos moksleivių informatikos olimpiadose, Jaunųjų programuotojų mokykloje, bendrojo lavinimo mokykloje, apžvelgiamos vertinimo metodų taikymo problemos. Algoritmavimas ir programavimas – artimos sąvokos, tačiau pirmoji vartojama norint labiau pabrėžti uždavinio algoritmo svarbą, jo suradimą, tuo tarpu programavimas kreipia dėmesį į algoritmo išraišką kompiuteriui tinkamu pavidalu. Aptariamas testavimas – pagrindinis algoritmų vertinimo būdas, taip pat kontrolinių testų sudarymo principai. Ypatingas dėmesys skiriamas uždavinių idėjos aprašymui ir algoritmavimo kultūrai.

Raktiniai žodžiai: programavimo mokymas, algoritmavimas, testavimas, vertinimas, informatikos olimpiados.

1. Įvadas

Lietuvoje (ir pasaulyje) reikia specialistų, giliai išmanančių kompiuterių, informatikos problemas. Tam pradedama rengtis bendrojo lavinimo mokykloje. Pripažįstama, kad rimčiausios kompiuterio sampratos žinios ir įgūdžiai įgyjami mokantis programavimo, algoritmavimo. Todėl veikia įvairios programavimo mokyklos, būreliai. Programuoti mokoma ir mokykliniame informacinių technologijų kurse.

Su algoritmavimu supažindinama bendrajame pagrindinės mokyklos kurse [1, 2]. Programavimo žinias galima gilinti pasirinkus išplėstinį modulį XI–XII klasėse [3].

Mokant algoritmavimo metodų ir programavimo pagrindinis dėmesys skiriamas programoms rašyti. Čia iš karto atsiranda vertinimo problema. Kokiais kriterijais remiantis vertinti?

Vertinimas – viena svarbiausių mokymo procesų dalių [8]. Vertinimas suvokiamas kaip informacijos rinkimas ir apibendrinimas tam, kad būtų galima daryti sprendimus, iš esmės gerinti mokymo procesą. Šiuos klausimus nagrinėja edukologai, galima būtų išvardinti daugybę studijų. 2004 m. Lietuvos Respublikos švietimo ir mokslo ministerija patvirtino dokumentą „Mokinių pažangos ir pasiekimų vertinimo samprata“ [11], kuriame aiškiai nubrėžtos bendrojo lavinimo nuostatos vertinimo atžvilgiu.

Šiame straipsnyje nagrinėsime specifinę sritį – algoritmų bei programų vertinimą. Daugiausiai dėmesio skirsime testams. Vertinant programas testų sąvoka visiškai skirtinga nuo priimtose švietime (kur kas platesnės). Programavime testu laikomas duomenų rinkinys, skirtas programai, algoritmui patikrinti.

Vertinimo problema opi ir informatikos olimpiadose – pasaulinėse ir nacionalinėse. Objektiviai įvertinti sunkesnio uždavinio sprendimą analizuojant programas teksta nėra lengva. Todėl pasirinktas automatinis testavimo būdas (Lietuvos olimpiadose –

nuo 1995 m.). Jis nuolat tobulinamas ir kito nuo paprasčiausių veiksmų automatizavimo iki ištisos tinklinės sistemos naudojimo. Nacionalinėse olimpiadose automatinis testavimo būdas derinamas su vizualiu programų vertinimu, kreipiamas dėmesys į algoritmo esmę, programavimo stilių bei kultūrą [5].

2. Algoritmų vertinimo būdai

Algoritmų vertinimas paprastai priskiriamas kriteriniam vertinimui – vertinama parinkus tam tikrus kriterijus, pavyzdžiui, ar programa teisingai veikia, įvertinama atlikus ją su parinktais pradinių duomenų ir rezultatų rinkiniais, vadinamaisiais testais. Galima būtų išskirti tris pagrindinius algoritmų vertinimo metodus: 1) verifikavimas, specifikavimas, 2) recenzavimas, aprašymas, 3) testavimas.

Algoritmavimo mokymo procese reikėtų aiškiai išskirti du aspektus: ar norima vertinti tik sukurtos programos teisingumą, ar siekiama aprėpti plačiau, visapusiškiau, įskaitant projektavimo ciklą.

Programos verifikavimas ar specifikavimas – tai tikslus programos teisingumo įrodymas. Tai sunkus metodas, praktiškai jis netaikomas programoms, kurios sudaromos mokantis programuoti.

Recenzavimas, sukurto algoritmo įvertinimas jį analizuojant, nurodant klaidas – tinkamas būdas mokinių mokymosi motyvacijai skatinti. Tačiau jis reikalauja individualaus darbo su mokiniais – tai brangu. Taip pat čia neišvengiama subjektyvumo. Aprašomasis vertinimas nepakeičiamas kai kurioms algoritmavimo mokymo dalims vertinti, pavyzdžiui, uždavinio sprendimo idėjai.

Programos kūrimo ciklas ilgas, susidedantis iš daugelio dalių. Ne visos jos vienodai vertinamos. Dažniausiai išskiriamos šios programos vertinimo dalys: 1) uždavinio sprendimo idėjos aprašas, 2) programos teisingumas, 3) algoritmavimo kultūra. Daugiausiai akcentuojama aiškiai pamatuojamoji dalis – testavimas.

3. Sprendimo idėjos vertinimas

Sprendimo idėja – tai trumpas, aiškus uždavinio sprendimo metodo aprašymas. Kartu tai yra vienareikšmiškas ir suprantamas uždavinio sprendimo algoritmo neformalus aprašas. Perskaičius aprašymą turi būti aišku, kaip bus sprendžiamas uždavinys. Apraše neturi būti vartojami jokie programos elementai (kintamųjų vardai, procedūrų bei funkcijų pavadinimai ir pan.). Galima pateikti matematinės formules, matematinius sprendimo būdus (pvz., duomenims vaizduoti grafo forma; lygčių sistemas užrašyti matematiniais žymenimis ir pan.).

Aprašas rašomas taisyklinga kalba. Vertinant atsižvelgiama į minties dėstymo nuoseklumą ir aiškumą, kalbos stilių. Aprašas teigiamai vertinamas, jeigu jį perskaičius tampa aiškus uždavinio sprendimo būdas bei įsitikinama, kad jis teisingas.

Lietuvos moksleivių informatikos olimpiadoje vertinant sprendimo idėją taškai skiriami:

- už idėjos aprašymą;
- už idėjos įgyvendinimą (skiriama, jeigu programa vertinama ne mažiau kaip puse jai skirtų taškų).

Idėjos aprašas vertinamas nulių taškų, jeigu:

- pateiktas klaidingas sprendimo būdas;
- mintys dėstomos nesuprantamai, painiai, neaiškiai.

Algoritmavimo uždavinių sprendimo idėjų vertinimas daro itin didelį poveikį besimokančiųjų mąstymui, padeda sisteminti mintis, jas glaustai ir aiškiai išreikšti.

4. Kontrolinių testų parinkimo principai

Testavimas laikomas objektyviausiu algoritmų vertinimo metodu. Objektyvumas pirmiausiai pasireiškia tuo, kad vertinant daug tą patį uždavinių sprendžiančių programų joms pateikiami tie patys testai, t. y. visų sprendimų vertinimo sąlygos yra vienodos. Tačiau pačių testų kūrimas turi subjektyvumo ir tai neišvengiama. Galima padidinti testų teisingumą laikantis tam tikrų testų sudarymo principų.

Labai sunku testais nustatyti programos teisingumą. Testais tik parodoma, kad programa gerai veikia tam tikroje duomenų aibėje. Kuo geriau, kruopščiau, pagal tam tikrus principus parenkami testai, tuo didesnei duomenų aibei programa duoda teisingus rezultatus.

Testai rengiami dvejopai: juos parenka pats programuotojas, norėdamas įsitikinti sukurtos programos semantiniu teisingumu, arba testus sukuria kiti žmonės, kurie vertina programą. Abiem atvejais testai turėtų tenkinti tuos pačius kriterijus, tačiau paprastai programos autorius testams kurti neskiria pakankamai dėmesio (arba neturi tam laiko, pavyzdžiui, olimpiadose).

5. Algoritmavimo kultūros vertinimas

Algoritmavimo kultūra susirūpinta, kai algoritmai bei programos tapo kolektyvinės kūrybos objektu: didesnėms programoms parašyti teko suburti didesnę programuotojų grupę, atsirado poreikis parašytus algoritmus tobulinti arba modifikuoti kitiems. Tada imta daugiau dėmesio skirti programų tekstams, susirūpinta jų aiškumu, vaizdumu, suprantamumu. Atsirado struktūrinis programavimas [7]. Buvo nustatyta, kad daugeliu atveju geru stiliumi parašytos programos pasižymi didesniu efektyvumu.

Lietuvoje algoritmavimo kultūrai daug dėmesio skiriama Jaunųjų programuotojų mokykloje [4, 6]. Čia išbandytos algoritmavimo kultūros mokymo bei vertinimo idėjos perėjo ir į algoritmavimo mokymą bendrojo lavinimo mokykloje, ir į informatikos olimpiadas [9, 10]. Paprastai informatikos olimpiadose algoritmavimo kultūra vertinama, jeigu programa surenka bent pusę taškų už testus.

Programa turėtų būti apiforminama laikantis programavimo technologijos principų: struktūrizuotai parenkamos tinkamos struktūros, prasminiai vardai, rašomi komentarai.

Vertinamas ir programavimo stilius – programos teksto išdėstymas, sakinių struktūra, komentavimas ir pan.

Vertinant programos tekstą atsižvelgiama, ar:

- programos struktūra atitinka reikalavimus (yra antraštė);
- visame tekste laikomasi vienodų ir aiškių teksto išdėstymo taisyklių;
- kintamųjų vardai ir kiti vardai visoje programoje yra mnemoniški (primena jais pažymėtus objektus), rašomi vienodu stiliumi;

- programa parengta laikantis struktūriškumo – uždavinio sprendimas suskaidytas į atskirų veiksmų grupes;
- gebama išvengti nereikalingų globaliųjų parametrų: procedūros bei funkcijos gauna duomenis ir perduoda rezultatus naudojantis parametrais;
- parinktos tinkamos duomenų struktūros; jos aprašomos programos teksto pradžioje ir galioja visoje programoje; nėra lokaliųjų duomenų struktūrų;
- laikomasi vaizdžių teksto skyrybos taisyklių: po skyrybos ženklų paliekamas tarpas, operacijų ženklai atskiriami tarpais, paliekami didesni tarpai tarp eilučių (tuščios eilutės) arba tarp ženklų grupių eilutėje ir pan.;
- komentuojamos esminės teksto vietos, veiksmų grupės, kintamieji, nusakoma procedūrų bei funkcijų paskirtis.

6. Išvados

Mokant programavimo, ypač bendrojo lavinimo mokykloje, tinkamas algoritmų vertinimo būdas būtų darbų recenzavimas, tačiau jis reikalauja daug sąnaudų. Todėl norint įvertinti mokinių žinias taikytinas programų testavimo metodus. Siekiant mokomųjų tikslų iš visų programos kūrimo ciklo dalių dažniausiai vertinama uždavinio sprendimo idėjos aprašas, programos teisingumas ir algoritmavimo kultūra.

Literatūra

1. J. Blonskis, V. Dagienė, Programavimo pagrindų mokymo vidurinėje ir aukštojoje mokykloje lyginamoji analizė, *Informacijos mokslai*, **26**, 23–28 (2003).
2. V. Dagienė, Programming-based solution of problems in informatics curricula, in: *Communications and Networking in Education: Learning in a Networked Society*, IFIP WG 3.1 and 3.5, Aulanko, Hämeenlinna, Finland, June 13–18 (1999), pp. 88–94.
3. V. Dagienė, J. Blonskis, Programavimo mokymas išplėstiniame informatikos kurse, *Liet. matem. rink.*, **42** (spec. nr.), 229–234 (2002).
4. V. Dagienė, V. Dagys, G. Grigas, Jaunųjų programuotojų mokyklos uždaviniai per 20 metų, *Informatika*, **1**(37), 45–62 (2001).
5. V. Dagienė, J. Skūpienė, Algoritmavimo uždavinių sprendimo metodų ir sudėtingumo analizė Lietuvos informatikos olimpiadose, *Liet. matem. rink.*, **43** (spec. nr.), 209–214 (2003).
6. V. Dagys, I. Stasiukaitytė, Kaip ir ko mokomasi Jaunųjų programuotojų mokykloje, in: *Kompiuterininkų dienos – 2003*, Žara, Vilnius (2003), pp. 69–76.
7. O.-J. Dahl, E.W. Dijkstra, C.A.R. Hoare, *Structured Programming*, Academic Press, London and New York (1972).
8. N.L. Gage, D.C. Berliner, *Pedagoginė psichologija*, Alma litera, Vilnius (1994).
9. G. Grigas, Informatikos olimpiadų darbų analizė programavimo stiliaus požiūriu, in: *Kompiuterininkų dienos '95*, Žara, Vilnius (1995), pp. 157–159.
10. G. Grigas, Investigation of the relationship between program correctness and programming style, *Informatika*, **6**(3), 265–276 (1995).
11. Mokinių pažangos ir pasiekimų vertinimo samprata. LR švietimo ir mokslo ministerija, 2004-02-25, įsakymas Nr. 256.
Adresas internete: <http://www.pedagogika.lt/samprata.pdf>

SUMMARY

V. Dagiėnė, V. Dagys. Difficulties in the evaluation of solving algorithms

The paper deals with the methods of algorithmical problems solving used in the Lithuanian Student Olympiads of Informatics, Young Programmers School, and comprehensive schools. Difficulties rising in the employment of the evaluation methods are discussed as well. The paper analyzes testing, which is the main method of algorithm evaluation, and the principles of the control tests drafting. The special attention is concentrated on the algorithm culture and the describing of the problems' ideas.

Keywords: programming teaching, algorithmization, testing, evaluation, olympiads of informatics.