

Algoritmų testavimas

Nelengva įsitikinti, ar parašytas algoritmas teisingas, t.y. ar jis daro tai, kas buvo suformuluota sąlygoje. Algoritmo teisingumu įsitikinama atlikus jį su tam tikrais kontroliniais duomenimis. Tokiu atveju sakoma, kad algoritmas testuojamas. Svarbu gerai sudaryti testus, kad jie apimtų kuo įvairesnių duomenų.

Kurdami bent kiek sudėtingesnę algoritmą, galime lengvai suklysti ir parašyti ne tuos veiksmus, kuriuos reikia atlikti. Dalį klaidų aptinkame vykdydami algoritmą kompiuteriu: kai kurias praneša kompiliatorius (jos vadinamos sintaksės klaidomis), pavyzdžiui, kai trūksta kabliataškio, neaprašytas kintamojo vardas, nėra skliausto. Tačiau būna daug tokių klaidų, kurias galima aptikti tik gerai įsigilinus į patį algoritmą. Pavyzdžiui, kai pradiniai duomenys gana dideli, gavus rezultatą - irgi nemažą skaičių, iš pirmo žvilgsnio sunku pasakyti, ar jis teisingas ar ne, - tenka šiek tiek paskaičiuoti.

Ypač lengva suklysti, kai yra ciklų. Todėl labai svarbu išmokti patikrinti algoritmus, rasti ir ištaisyti klaidas.

Tokiu atveju, kai vartojamas kintamasis, kuris reikšmės neturi, sakoma, kad **kintamojo reikšmė neapibrėžta** (arba neapibrėžtas kintamasis). Tai viena iš dažniausiai pasitaikančių klaidų.

Kita dažnai pasitaikanti klaida yra niekad nesibaigiantis, arba amžinasis, ciklas.

“Apsiavė ji tomis korpėmis, eina, mina, brūžina į plytas, į akmenis, kur tik pagriebsi, bet korpės storos, kietos, visiškai nedyla. Minsi neminsi - jų visam amžiui užteks”. Tai jums gerai žinomos pasakos “Eglė žalčių karalienė” ištrauka: kurpių nešiojimas būtų amžinasis ciklas. Tik kalvis įsikišo ir nutraukė šį ciklą - pagruzdino kurpes žaizdre, ir jos susidėvėjo.

Amžinasis ciklas - dažnai pasitaikanti klaida, ypač sudėtingesniuose cikluose. Todėl užrašius ciklą, būtinai reikia įsitikinti, ar jis nėra amžinasis.

Kad ciklas nebūtų amžinasis, reikia, kad būtų keičiamos bent vieno kintamojo, įeinančio į ciklo sąlygą, reikšmės.

Nors šį pavyzdį galima išspręsti ir be kompiuterio (patys matome, kokie skaitmenys sudaro skaičių), tačiau algoritme padaryta klaida būdinga daugeliui sudėtingesnių uždavinių. Be to, šis algoritmas gali būti reikalingas sprendžiant skaitinius uždavinius, kai skaičių (pradinį duomenį) pateikia ne žmogus, o kompiuteris.

Būna ir kitokių klaidų. Daugelį jų galime rasti patys, tikrindami algoritmą. Tam parenkame paprastesnes pradinių duomenų reikšmes ir atliekame algoritmo veiksmus taip, kaip juos atliktų kompiuteris. Popieriaus lape surašome programoje vartojamus kintamųjų vardus ir jų pradines reikšmes. Kintamajam priskyre naują reikšmę, ankstesniąją užbraukiame, ir jos vietoje rašome naują. Jeigu taip apskaičiuoti rezultatai nesutampa su tais, kurie turėtų būti pagal uždavinio sąlygą, darome išvadą, kad algoritme yra klaida.

Tikrinimui parinktos pradinės kintamųjų reikšmės vadinamos **kontroliniais duomenimis**. Jie turi būti tinkami, būdingi tam algoritmui. Pavyzdžiui, kai algoritme yra sąlyginių sakinių, reikia parinkti tokius kontrolinius duomenis, kad bent po kartą tektų atlikti kiekvieną sąlyginio sakinio šaką. Tikrinant

Parengė ITMM Artūras Šakalys

algoritmus, kuriuose yra ciklą, kontrolinius duomenis pravartu pasirinkti tokius, kad būtų galima ciklo neatlikti nė karto, atlikti jį vieną kartą, atlikti keletą kartų.