Labirintas

Uždavinys – rasti išėjimą iš labirinto – yra klasikinis. Jam yra žinomi ir klasikiniai sprendimo metodai. Labirinto uždavinys dažniausiai sprendžiamas vienu iš šių būdų – grįžimo (arba paieškos į gylį) metodu, bangos algoritmu, paieškos į plotį metodu.

Sprendimas, atrodo, gana akivaizdus. Pirmiausia reikia apeiti visą labirintą visomis kryptimis (t. y. įsiminti patį labirintą), o tuomet pritaikyti, pavyzdžiui, bangos metodą. Stop! Juk apeidami labirintą visomis kryptimis, tikrai prieisime prie išėjimo. Kam dar betaikyti kokį nors algoritmą?

Mūsų uždavinys nuo klasikinio šiek tiek skiriasi: neprašoma rasti trumpiausio išėjimo iš labirinto, o prašoma tiesiog išeiti iš jo. Pats labirintas nėra duotas. Nepasakyta net, iš kurios pusės yra įėjimas į labirintą.

Bangos ar paieškos į plotį algoritmą galima taikyti ir neapėjus viso labirinto. Tačiau jie nagrinėja ne gretimus labirinto langelius, o langelius, kurie vienodai nutolę nuo labirinto įėjimo. Reikia nepamiršti, kad šiame uždavinyje vaikštoma labirintu ir norint pakliūti nuo vieno tokio (vienodai nutolusio) langelio prie kito, gali tekti sukarti nemažą atstumą.

Paprasčiausia taikyti grįžimo metodą. Šiuo atveju jį galima pavadinti *Kairiosios rankos taisykle* – visą laiką einame laikydamiesi kairiosios sienos ir išėjimą būtinai pasieksime. Lygiai taip pat galime pasirinkti ir dešiniąją sieną ir eiti pagal ją. Taip vaikštant nereikia įsiminti nei ėjimų, nei paties labirinto.

Pradžioje reikia nustatyti, iš kurios pusės yra įėjimas į labirintą. Tai nesunku padaryti: reikia kreiptis į funkcijas š*iaurė, pietūs, rytai, vakarai*. Bus leidžiama eiti tik viena iš šių krypčių. Mat sąlygoje pasakyta, kad įėjimą iš abiejų pusių riboja sienos. Taigi akivaizdu, kad į dvi puses negalėsime eiti. Grįžti pro įėjimą taip pat negalima. Šitai nepasakyta sąlygoje, tačiau tai galima nustatyti išbandžius modulį.

Tarkime, kad eiti leidžiama į šiaurę. Tai reiškia, kad eidami pirmyn eisime į šiaurę. Jei beeidami labirintu nuspręsime, kad norime eiti į vakarus, teks eiti kairėn. Kai kreipiamės į procedūrą *kairėn*, prieš paeidami pasisukame į kairę. T. y. atlikę šį žingsnį, žiūrime vakarų kryptimi. Vaikštant labirintu svarbu sekti, kuria kryptimi žiūrime, t. y. kokia kryptimi (šiaurės, pietų, rytų, vakarų) eisime darydami žingsnį *pirmyn*.

Taigi iš labirinto išeinama šitaip: jei galima, einama į kairę, jei ne, sužinoma, ar galima eiti pirmyn. Jei galima, ten einama, jei ne – bandoma eiti dešinėn, jei ir ten negalima – tada einama atgal.

Prieš pateikiant sprendimą reikia pakomentuoti porą dalykų. Modulyje pateikta funkcija *dydis*. Ji sąlygoje buvo pateikta sąmoningai, nors geram uždavinio sprendimui ji visiškai nereikalinga. Antra, sąlygoje prašoma, kad ėjimų skaičius labirinte būtų kuo mažesnis. Tai užuomina į sprendimą: jei bandysite taikyti bangos ar paieškos į plotį algoritmą, rasite trumpiausią kelią, bet ėjimų padarysite labai daug...

```
program labirintas;
  uses labir;
  type kryptys = (kair, pirm, deš, atg);
       šalys = (vak, šiaur, ryt, piet);

  function pred_š (šalis: šalys): šalys;
  begin
    if šalis = vak
       then pred_š := piet
       else pred_š := pred (šalis);
  end; { pred_š }

  function succ_š (šalis: šalys): šalys;
  begin
    if šalis = piet
       then succ_š := vak
       else succ_š := succ (šalis)
  end; { succ_š }

  function galima (kryptis: kryptys; šalis: šalys)
                                        : boolean;
  { nustato, ar galima eiti nurodyta kryptimi, jei žmogus veidu }
  { atsisukęs kryptimi šalis }
  begin
    case kryptis of
      kair: šalis := pred_š (šalis);
      deš: šalis := succ_š (šalis);
      atg: šalis := pred_š (pred_š (šalis))
    end;
    case šalis of
      vak: galima := vakarai;
      šiaur: galima := šiaurė;
      piet: galima := pietus;
      ryt: galima := rytai
```

```
    end;
  end;  { galima }

  procedure eiti (kryptis: kryptys; var šalis: šalys);
  { atliekamas ėjimas nurodyta kryptimi }
  { rezultatas – pasaulio šalis, į kurią lieka atsisukęs žmogus po ėjimo }
  begin
    case kryptis of
      kair: kairėn;
      deš: dešinėn;
      atg: atgal;
      pirm: pirmyn;
    end;
    case kryptis of
      kair: šalis := pred_š (šalis);
      des: šalis := succ_š (šalis);
      atg: šalis := pred_š (pred_š (šalis))
    end
  end;  { eiti }
  var kryptis: kryptys;
      paeita: boolean;
      šalis: šalys;
begin
  { nustatoma kuria kryptimi žiūri žmogus stovintis labirinto }
  { įėjime }
  if šiaurė
     then šalis := šiaur
     else if pietūs
             then šalis := piet
             else if rytai
                     then šalis := ryt
                     else šalis := vak;
  { kol neišėjome iš labirinto }
  while true do
    begin
      kryptis := kair;  { bandysime eiti į kairę }
      paeita := false;
      while not paeita do
        if galima (kryptis, šalis)  { jei galima paeiti šia kryptimi }
           then begin
                  eiti (kryptis, šalis);
                  paeita := true
                end
           else kryptis := succ (kryptis)
    end
end.
```

Olimpiados dalyviams modulio LABIR tekstas nebuvo pateiktas, tik sukompiliuotas modulis. Norintys panagrinėti modulio tekstą, gali jį rasti diskelyje. Diskelyje galite rasti ir kitą modulį LAB_DEM. Šis modulis piešia ekrane labirintą ir demonstruoja kaip jame atliekami jūsų ėjimai. Jį galite naudoti, jei labirintas nedidelis, t. y. dar telpa ekrane.