



Diskelių paskirstymas

Tai gyvenimiškas uždavinys, kurį iš tikrųjų sprendė Matematikos ir Informatikos instituto vadovybė, skirstydama gautus diskelius instituto padaliniams. Jo ir formuluotė nėra labai griežta – aiškiai nepasakyta, kas argumentai ir kas rezultatai. Tai padaryta neatsitiktinai: norėta, kad mokiniai patys pagalvotų ir užbaigtų uždavinio formuluotę.

Paprasčiausias algoritmas – dalyti po vieną diskelį pradedant nuo to, kuris daugiausiai prašė. Kai visi bus gavę po vieną diskelį, vėl imti dalyti po vieną pradedant nuo daugiausiai prašiusio, bet tikrinant, ar nėra tokių pirkėjų, kuriems jau užtenka diskelių. Taip kartoti tol, kol pasibaigs diskeliai.

Šitaip dalydami po vieną diskelį, palaipsniui pasieksime ribinį skaičių C . Bus tenkinama pirmoji sąlyga. Dalijant paskutinį kartą, visiems diskelių nebeužteks, todėl kai kas gaus $C + 1$ diskelių, o kai kam paskutinio diskelio nebeliks ir bus gavęs C diskelių – tenkinama antroji sąlyga. Kadangi pradedama dalyti nuo tų, kurie daugiau prašė, tai tenkinama trečioji sąlyga. Paskutinė sąlyga bus patenkinta savaime: dalijama tol, kol yra.

Diskelius galima išdalyti greičiau, jeigu dalytume ne po vieną, o po daugiau. Pirmoji porcija gali būti $k = d \text{ div } n$ diskelių. Tik reikia patikrinti, ar $k > D_i$. Jei taip, tai i -ajam pirkėjui duoti ne visą porciją (k), o tik tiek, kiek jis prašė, t. y. D_i diskelių. Kai diskelių liks mažiau, negu dar nepatenkintų pirkėjų (t. y. negavusių tiek, kiek prašė) reikia dalyti po vieną pradedant nuo tų, kurie prašė daugiausiai.

Yra dar efektyvesnis algoritmas. Pirmiausia apskaičiuokime vidutinį diskelių skaičių C , kurį galima parduoti aktyviems (t. y. dar norintiems pirkti) pirkėjams. Pradžioje aktyvūs visi pirkėjai. Pradedama dalyti nuo pirkėjų, pageidavusių mažiausiai diskelių. Jeigu pirkėjas prašė daugiau diskelių negu C , tai jam duodama C diskelių, priešingu atveju – tiek, kiek prašė. Nesunku įsitikinti, kad su pastaruoju pirkėju jau bus galutinai atsiskaityta, todėl perskaičiuojamas vidutinis diskelių skaičius C likusiems aktyviems pirkėjams ir kartojami tie patys veiksmai.

Belieka užrašyti patį algoritmą.

Pradiniai algoritmo duomenys – pirkėjų skaičius, bendras gautų diskelių skaičius ir atskirų pirkėjų pageidaujami diskelių skaičiai.

Rezultatai – kiekvieno pirkėjo įsigytų diskelių skaičius bei skaičius C . Sąlyga neprašo surasti šio skaičiaus, bet jį įtraukėme į rezultatus. Žinant šį skaičių, lengviau įsitikinti, ar algoritmo rezultatas toks, kokio reikalauja uždavinio formuluotė.



```
const P_MAX      = 100; { didžiausias pirkėjų skaičius }
type duomenys    = record
                    nr,                { pirkėjo numeris }
                    prašo,            { kiek prašo }
                    gauna: integer    { kiek gauna }
                end;
lentelė = array [1..P_MAX] of duomenys;

procedure diskeliai (d, { gautų diskelių skaičius }
                    p: integer; { pirkėjų skaičius }
                    var C: integer; { skaičius C }
                    var dsk: lentelė);

var i, j: integer;
    tarp: duomenys;
begin
    for i := 1 to p do
        begin
            dsk[i].nr := i;
            dsk[i].gauna := 0
        end;
    { pagal prašomų diskelių skaičius pirkėjų masyvas surikiuojamas didėjimo tvarka }
    for i := 1 to p-1 do
        for j := 1 to p-i do
            if dsk[j].prašo > dsk[j+1].prašo
            then begin
                tarp := dsk[j];
                dsk[j] := dsk[j+1];
                dsk[j+1] := tarp
            end;
    { dalijami diskeliai }
    for i := 1 to p do
        begin
            C := d div (p - i + 1);
            if dsk[i].prašo <= C
            then begin
                dsk[i].gauna := dsk[i].prašo;
                d := d - dsk[i].prašo
            end
            else begin
                dsk[i].gauna := C;
                d := d - C
            end
        end;
    end;
    C := C - 1
end;
```