



Kaladėlių perdėliojimas

Perrinkti visus perdėliojimo variantus – nerealu. Jų yra $1000!$, t. y. skaičius, turintis ne mažiau 2000 skaitmenų.

Pabandykite įrodyti, kad kaladėlės pavyks perdėlioti tada ir tik tada, jei nė viena raidė nebus užrašyta daugiau kaip ant pusės kaladėlių. Jei kuri nors raidė užrašyta daugiau kaip 500 kartų, perdėlioti nepavyks.

Nesunkiai galite įsitikinti: jei kaladėlės su raide, kuri užrašyta 500 kartų, sukeisime su likusiomis kaladėlėmis, tai sąlyga bus tenkinama. Jei raidė pasitaiko daugiau kartų, pvz., 501, tuomet 499 poras kaladėlių galima sukeisti, o paskutinė pora liks su vienodomis raidėmis ir nebebus su kuo jas sukeisti.

Lieka įrodyti, kad jei bet kuri raidė užrašyta ne daugiau kaip ant pusės kaladėlių, tai kaladėlės perstatyti galima.

Pradėkime nuo paprastesnio atvejo. Tarkime, kad kaladėlės surikiuotos pagal abėcėlę. Sunumeruokime jas. Pavyzdžiui, duota šitokia kaladėlių eilė (tolesniam nagrinėjimui pakaks ir dešimties kaladėlių):

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|
| A | A | A | B | C | C | C | C | N | N |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Dabar perkelsime kaladėles cikliška per 5 vietas ($10 \text{ div } 2 = 5$) į priekį. Gausime šitokią kaladėlių eilę:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|
| C | C | C | N | N | A | A | A | B | C |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Kadangi nė viena raidė neužrašyta daugiau kaip ant penkių (pusės) kaladėlių, tai pirma netgi iš daugiausiai pakartotų raidžių peršoks per visas savas raides, o paskutinė „neužšoks“ ant savos. Taigi nė viena raidė neliks senoje vietoje.

Tarkime, duota k kaladėlių, o dažniausiai pasikartojanti raidė užrašyta ant d kaladėlių. Bendru atveju raidė neliks senoje vietoje tik tada, kai poslinkis bus iš sveikųjų skaičių intervalo $[d; k \text{ div } 2]$. Jei kaladėlių perdėlioti negalima, šis intervalas tuščias.

Jei kaladėlės perdėlioti galima, i -oji kaladėlė turės būti perkelta per $k \text{ div } 2$ pozicijų, t. y. į poziciją $i + k \text{ div } 2$. Kadangi kaladėlės perkeliamos ratu, t. y. po k -osios pozicijos eis 1-oji, kaladėlę reikia perstatyti į $(i + k \text{ div } 2 - 1) \bmod k + 1$ vietą. Papildomi vienetai atsiranda todėl kad kaladėlės numeruojamos nuo 1 iki k , o dalybos iš k liekana yra iš intervalo $[0..k-1]$.



Pateiktas įrodymas tiktų ir nesurikiuotų abėcėlės tvarka kaladėlių atveju, nes kaladėles galima sunumeruoti taip, kad jų numeriai eitų ne pagal duotą kaladėlių eilę, o pagal abėcėlės raides:

Pavyzdžiui, jeigu kaladėlės buvo išdėstytos šitaip:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | B | C | N | A | A | N | C | C | C |
|---|---|---|---|---|---|---|---|---|---|

tai pirmiausia sunumeruojame visas raides A, po to – raides B ir t. t. Gauname šitokią kaladėlių numeraciją:

| | | | | | | | | | |
|---|---|---|---|---|---|----|---|---|---|
| A | B | C | N | A | A | N | C | C | C |
| 1 | 4 | 5 | 9 | 2 | 3 | 10 | 6 | 7 | 8 |

Tada i -uoju numeriu pažymėtą kaladėlę tereikia perkelti į $(i + k \text{ div } 2 - 1) \bmod k + 1$ numeriu pažymėtos kaladėlės vietą.

Remdamiesi įrodymu, užrašysime uždavinio sprendimą:

Pirmiausia suskaičiuosime, kiek kokių raidžių užrašyta ant kaladėlių. Jei nors viena raidė sutinkama daugiau nei 500 kartų (arba bendresniu atveju daugiau nei $k \text{ div } 2$ kartų, čia k – kaladėlių skaičius), perdėlioti negalima.

Priešingu atveju visas kaladėles sunumeruojame bei sudarome pagalbinę lentelę, kuri padėtų surasti turimu numeriu pažymėtą kaladėlę.

Toliau pagal anksčiau pateiktą formulę perskaičiuojame kiekvienos kaladėlės numerį ir, pasinaudodami pagalbine lentele, perkeliame kaladėlę į naują vietą.

Pavyzdžiui:

Pradiniai duomenys

Pradinė numeracija

Nauja numeracija

Rezultatai

| | | | | | | | | | |
|---|---|---|---|---|---|----|----|---|---|
| A | B | C | N | A | A | N | C | C | C |
| 1 | 4 | 5 | 9 | 2 | 3 | 10 | 6 | 7 | 8 |
| 5 | 8 | 9 | 3 | 6 | 7 | 4 | 10 | 1 | 2 |
| C | N | A | C | C | N | C | A | A | B |



```
const K = 1000;          { kaladėlių skaičius }
type kalad = array [1..K] of char;
      numeracija = array [1..K] of 0..K;

procedure perdėlioti (a: kalad;          { pradinė eilė }
                     var b: kalad; { perdėliota eilė }
                     var galima: boolean);
{ nustatoma, ar galima kaladėles perdėlioti taip, kad nė viena raidė nepasiliktų }
{ savo vietoje ir jei galima – kaladėlės perdėliojamos }
  var raid: array ['A'..'Z'] of 0..K; { raidžių pasikartojimų skaičius }
      numer,          { raidžių numeracija }
      kur: numeracija; { kur yra tam tikru numeriu pažymėta kaladėlė }
      nau_num,        { perkeltos kaladėlės numeris }
      i, nm: integer;
      j: char;
begin
  for j := 'A' to 'Z' do
    raid[j] := 0;
  for i := 1 to K do { kiek kartų pasikartoja kiekviena raidė }
    raid[a[i]] := raid[a[i]] + 1;
  galima := true;
  for j := 'A' to 'Z' do
    galima := galima and (raid[j] <= K div 2);
  { jei kaladėlių perdėlioti negalima, darbas baigiamas }
  if not galima then exit;
  { jei galima – perdėliojama }
  { sunumeruojamos kaladėlės ir sudaroma lentelė, padedanti rasti sunumeruotas kaladėles }
}
nm := 0;
for j := 'A' to 'Z' do
  for i := 1 to K do
    if a[i] = j
    then begin
      nm := nm + 1;
      numer[i] := nm;
      kur[nm] := i
    end;
  { perskaičiuojami kaladėlių numeriai ir perdėliojamos kaladėlės }
  for i := 1 to K do
    begin
      nau_num := (numer[i] + K div 2 - 1) mod K + 1;
      { i-osios kaladėlės naujas numeris }
      b[kur[nau_num]] := a[i]
    end
  end;
end;
```