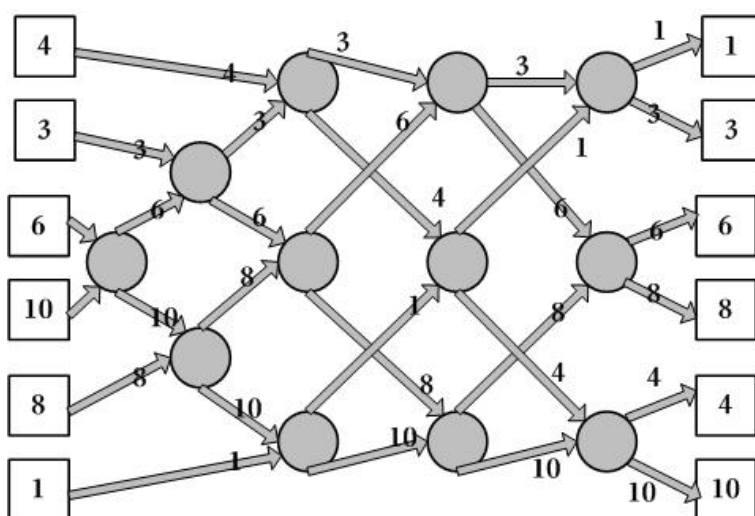




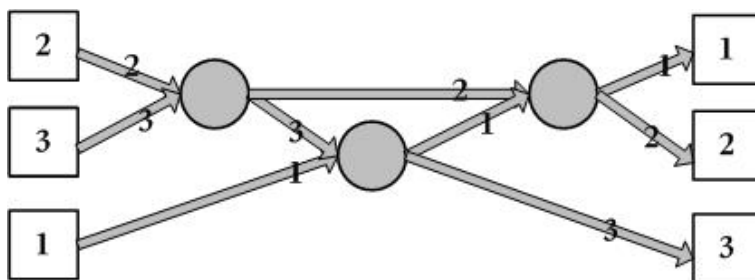
Uždavinių sprendimai

Rikiavimo tinklas (uždavinys VIII-XII klasėms). Šiame uždavinyje mokiniai supažindinami su nauja sąvoka — rikiavimo tinklais. Norint išspręsti uždavinį reikia išanalizuoti pateiktus tinklų pavyzdžius ir suvokti, kaip jie veikia.

Nekorektiškai rikiuoja pirmasis tinklas. Seką, kuri būtų rikiuojama nekorektiškai atrasti nesunku, nes šis tinklas daugiau nei 70% atvejų rikiuoja netinkamai. Tereikia išbandyti keletą pavyzdžių. Žemiau pateiktas nekorektiškai rikiuojamos sekos 4 3 6 10 8 1 pavyzdys.



Vyresniems papildomai reikia sukonstruoti tinklą, rikiuojantį seką iš trijų elementų. Žemiau pateiktas tokio tinklo pavyzdys. Sekos elementų nedaug ir paanalizavus tinklą nesunku įsitikinti, kad jis rikiuoja korektiškai. Tinklas yra simetriškas, tad ir apsuktasis tinklas korektiškai rikiuos sekas iš trijų elementų.



Informatiškas laikas (uždavinys VIII-IX klasėms). Šį uždavinį galima spręsti simuliuojant laiko tėkmę ir „sukant“ laikrodį. Skaičiuojant praėjusias sekundes reikia kaskart patikrinti, ar dar nepasiektas informatiškas laikas.

Galimas efektyvesnis sprendimas — iš eilės generuoti visus „informatiškus“ laikus, kol bus randamas mažiausias, viršijantis duotąjį laiką. Iš simbolių 0 ir 1 galime sudaryti 2^6 skirtingų „informatiškų“ laikų, o taip pat reikia nepamiršti atvejo, kai informatiško laiko tenka laukti iki kitos paros.



Pateiksime pirmo aprašyto sprendimo būdo pseudokodą. Antrąjį sprendimą rasite kartu su aprašymu pateiktame pakete.

```
1  read  $v, m, s$                 ▷ valandos, minutės, sekundės
2   $B \leftarrow \{0, 1, 10, 11\}$     ▷ informatiškų laikų aibė
3   $praėjo \leftarrow 0$             ▷ kiek sekundžių jau prasukome
4  ▷ ciklą vykdome, tik jei laikas dar neinformatiškas
5  while not ( $v \in B$  and  $m \in B$  and  $s \in B$ )
6      do  $praėjo \leftarrow praėjo + 1$ 
7           $s \leftarrow s + 1$ 
8          if  $s = 60$                 ▷ nauja minutė
9              then  $s \leftarrow 0$ 
10              $m \leftarrow m + 1$ 
11             if  $m = 60$             ▷ nauja valanda
12                 then  $m \leftarrow 0$ 
13                  $v \leftarrow v + 1$ 
14             if  $v = 60$             ▷ nauja para
15                 then  $v \leftarrow 0$ 
16  print  $praėjo$ 
```

Petras (uždavinys VIII-XII klasėms). Sprendžiant šį uždavinį reikia panaudoti vienmatį loginį masyvą. Jame skaitant duomenis pažymima, kurie miestelėnai yra klubo nariai. Tolesnių duomenų (t. y. kandidatų ir juos rekomenduojančių asmenų) įsiminti nebūtina, kadangi perskaičius informaciją apie kandidatą gali būti iš karto patikrinama, ar rekomenduojantys asmenys yra klubo nariai. Jei taip — kandidatas priimamas, t. y. naujų klubo narių skaičius padidinamas vienetu.

Žemiau pateikiamas algoritmo pseudokodas.

```
1  read  $N, M, K$     ▷ miestelėnų, narių ir kandidatų skaičius
2  for  $i \leftarrow 1$  to  $M$ 
3      do
4          read  $a$                 ▷  $i$ -asis narys
5           $narys[a] \leftarrow \text{TRUE}$  ▷ pažymime, kad a-asis miestelėnas yra narys
6   $priimta \leftarrow 0$             ▷ inicializuojame priimtų narių skaičių
7  for  $i \leftarrow 1$  to  $K$         ▷ tikriname kandidatus
8      do
9          read  $kand, r_1, r_2$ 
10         if  $narys[r_1]$  and  $narys[r_2]$ 
11             then  $priimta \leftarrow priimta + 1$ 
12  print  $priimta$ 
```

Čempionatas (uždavinys X-XII klasėms). Čempionato simuliacijai susikurkime tris sveikųjų skaičių masyvus.

Komandų pajėgumus surašykime į vieną masyvą. Kitą masyvą, kuriame modeliuosime komandų tarpusavio išsidėstymą, nuosekliai užpildome komandų numeriais nuo 1 iki N :



Lietuvos mokinių informatikos olimpiada

Miesto etapas • 2010 m. gruodžio 10 d.

Pajėgumai:

2	4	7	3	8	1	5	6
---	---	---	---	---	---	---	---

Komandos:

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Simuliuodami pirmą čempionato etapą, į laikino masyvo pirmąją pusę nuosekliai sudedame visų porų nugalėtojus, o į antrąją – pralaimėtojus, t. y. i -osios poros nugalėtoją dedame į i -ąją vietą, o pralaimėtoją – į $(\frac{N}{2} + i)$ -ąją.

Laikinas:

2	3	5	8	1	4	6	7
---	---	---	---	---	---	---	---

Baigus simuliuoti pirmą etapą, masyvo, kuriame saugomas komandų išsidėstymas, turinį pakeičiame laikinojo masyvo turiniu, ir, uždavinį padaliname į du smulkesnius.

Antrąjį etapą simuliuojame atskirai taikydami tą patį algoritmą pirmame etape laimėjusių ir pirmame etape pralaimėjusių komandų grupėms. Taigi jį atliekame su komandomis, kurių numeriai yra pirmuosiuose $\frac{N}{2}$ masyvo elementuose, bei su komandomis paskutiniuosiuose $\frac{N}{2}$ masyvo elementuose.

Simuliavimą ir dalinimą tęsiame tol, kol padalintos dalys tampa vienetinės – šitaip masyve gauname komandų numerių rikiuotę, kurios prašo sąlyga.

Po 2 etapų:

3	5	2	8	4	7	1	6
---	---	---	---	---	---	---	---

Po 3 etapų:

5	3	8	2	7	4	1	6
---	---	---	---	---	---	---	---

Toliau pateiktame algoritmo pseudokode šis dalijimas ir komandų pozicijų perkėlinėjimas realizuojamas įvedant šuolio savoką: nagrinėjamos vienodą pergalių skaičių turinčios komandos sudaro $2 \cdot \text{šuo}lis$ ilgio sekas, i -osios tos sekos poros nugalėtojas keliauja į i -ąją tos sekos vietą, o pralaimėtojas „nušoka“ į vietą $i + \text{šuo}lis$. Kiekviename etape $\text{šuo}lis$ mažėja pusiau, kol pasiekia nulį.



```
1  read  $N$                                 ▷ komandų skaičius
2  for  $i \leftarrow 1$  to  $N$ 
3      do
4          read  $pajėgumai[i]$  ▷  $i$ -osios komandos pajėgumas
5           $komandos[i] \leftarrow i$  ▷ inicializuojame pradinę komandų rikiuotę
6   $šuo\text{lis} \leftarrow N \text{ div } 2$           ▷ pirmame etape naudojame visą masivą, o „šoka“ per pusę jo ilgio
7  while  $šuo\text{lis} > 0$                       ▷ simuliuojame etapą
8      do
9           $i \leftarrow 1$ 
10         while  $i < N$                     ▷ komanda  $i$  žaidžia su komanda  $i + 1$ 
11             do
12                 if  $pajėgumai[komandos[i]] > pajėgumai[komandos[i + 1]]$ 
13                     then
14                          $laimėjo \leftarrow komandos[i]$ 
15                          $pralaimėjo \leftarrow komandos[i + 1]$ 
16                     else
17                          $laimėjo \leftarrow komandos[i + 1]$ 
18                          $pralaimėjo \leftarrow komandos[i]$ 
19                  $laikinas[1 + i \text{ div } 2 + (i \text{ div } (2 \cdot šuo\text{lis})) \cdot šuo\text{lis}] \leftarrow laimėjo$ 
20                  $laikinas[1 + i \text{ div } 2 + (i \text{ div } (2 \cdot šuo\text{lis}) + 1) \cdot šuo\text{lis}] \leftarrow pralaimėjo$ 
21                  $i \leftarrow i + 2$ 
22          $šuo\text{lis} \leftarrow šuo\text{lis} \text{ div } 2$ 
23          $komandos \leftarrow laikinas$ 
24  for  $i \leftarrow 1$  to  $N$ 
25      do print  $komandos[i]$ 
```

Pateikto algoritmo sudėtingumas $O(N \log N)$.