

**LATVIJAS 19. INFORMĀTIKAS OLIMPIĀDES III POSMA
UZDEVUMU APSKATS
Pirmā diena (2006.gada 20.marts)**



| Uzdevuma nosaukums: | IEVĀRĪJUMA BURCIŅAS | SKAITĻU ROTĒŠANA | VIRKŅU SUMMĒŠANA |
|--|---|---|---|
| Ievaddatu faila nosaukums: | burcinas.dat | rotesana.dat | virknas.dat |
| Izvaddatu faila nosaukums: | burcinas.rez | rotesana.rez | virknas.rez |
| Izpildes laika ierobežojums vienam testam (laiks tiek mērīts uz testēšanas servera): | 1 sekunde | 1 sekunde | 2 sekundes |
| Atmiņas ierobežojums: | 64MB | 64MB | 256MB |
| Maksimāli iespējamais punktu skaits par uzdevumu: | 100 | 100 | 100 |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā PASCAL : | { task: burcinas lang: pascal } | { task: rotesana lang: pascal } | { task: virknas lang: pascal } |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā C : | /* task: burcinas lang: c */ | /* task: rotesana lang: c */ | /* task: virknas lang: c */ |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā C++ : | /* task: burcinas lang: c++ */ | /* task: rotesana lang: c++ */ | /* task: virknas lang: c++ */ |
| Nosacījums, lai testēšanas serveris atzītu programmu par derīgu testēšanai: | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajam piemēram | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem |

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili atrodas tekošajā katalogā) un tieši tā, kā norādīts uzdevuma formulējumā (**ar mazajiem burtiem**)!

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL: FreePascal (versija 1.0.10) ar parametriem -So -Sg -O2 -Xs

Valodai C/C++: GNU C/C++ (versija 3.3.6) ar parametriem -O2 -static -lm

Jautājumus par uzdevumu formulējumiem centieties uzdot sacensību pirmās stundas laikā!

Katra uzdevuma labāko programmas kodu pēc nosūtīšanas uz serveri saglabājiēt arī darba datorā un neizdzēsiet pēc sacensību beigām!

LATVIJAS 19. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Pirmā diena (2006.gada 20.marts)



1. "IEVĀRĪJUMA BURCIŅAS"

Toma vecmāmiņai ir pagrabs, kurā ir garš plaukts, uz kura var salikt daudz ievārījuma burciņu. Viens plaukta gals atrodas tuvu pagraba durvīm, bet otrs iestiepjas tālu pagrabā. Uz plaukta, sākot ar tālāko plaukta galu, ar skaitļiem no 1 līdz 2147483647 pēc kārtas ir sanumurētas burciņām paredzētās vietas. Katrā vietā var novietot vienu burciņu. Vecmāmiņa ir noteikusi, ka plaukts jāsāk aizpildīt, sākot ar tālo galu - t.i., tikko vecmāmiņa ir savārījusi kārtējo ievārījuma porciju, tā Tomam jānododas uz pagrabu un burciņas jānovieto uz plaukta, sākot ar vistālāko galu, neizlaižot nevienu brīvo vietu. Katrs ievārījuma veids tiek apzīmēts ar vienu lielo latīņu alfabēta burtu - zemeņu ar "Z", plūmju ar "P", aveņu ar "A", utt. Viens un tas pats ievārījuma veids var tikt vārīts vairākkārt.

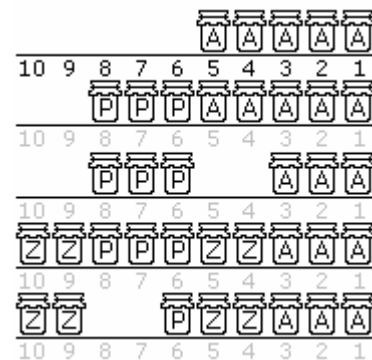
Kad sabrauc ciemiņi, vecmāmiņa cienā tos ar ievārījumu un Toma uzdevums ir no pagraba uznest noteiktu skaitu noteikta veida ievārījuma burciņu. Tā kā vecmāmiņa nav noteikusi, kādā secībā burciņas no pagraba jāņem, tad Toms vienmēr ņem pēc kārtas tās noteiktā ievārījuma veida burciņas, kas atrodas vistuvāk durvīm.

Ievārījuma vārīšana (kas izraisa burciņu ienešanu pagrabā) un ciemiņu ciemošanās (kas izraisa burciņu iznešanu no pagraba) var notikt jauktā secībā.

Ja pēc kārtas tiek veiktas šādas darbības:

- 1) ienestas piecas aveņu ievārījuma burciņas,
- 2) ienestas trīs plūmju ievārījuma burciņas,
- 3) iznestas divas aveņu ievārījuma burciņas,
- 4) ienestas četras zemeņu ievārījuma burciņas,
- 5) iznestas divas plūmju ievārījuma burciņas,

tad plaukta pirmo desmit vietu aizpildījums mainās tā, kā redzams zīmējumā.



Dažreiz vecmāmiņa pēkšņi grib uzzināt, kāda ievārījuma veida burciņa atrodas noteiktā plaukta vietā. Tā kā Tomam nepatīk iet uz pagrabu tikai tāpēc, lai apmierinātu vecmāmiņas ziņkārtību, viņam ir nepieciešama datorprogramma, kas nosaka, kāda ievārījuma veida burciņa ir uz plaukta vietā ar noteiktu numuru. Uzrakstiet Tomam šādu datorprogrammu!

Ievaddati

Teksta faila `burcinās.dat` pirmajā rindā dots naturāls skaitlis - veikto darbību skaits $N(1 \leq N \leq 10^5)$.

Katrā no nākošajām N faila rindām dots vienas darbības apraksts vienā no trim veidiem:

- 1) `<burts><tukšumsimbols><naturāls skaitlis>`

Apraksta burciņu ienešanu pagrabā. Burts norāda ievārījuma veidu un skaitlis - nesamo burciņu skaitu.

- 2) `<burts><tukšumsimbols><vesels negatīvs skaitlis>`

Apraksta burciņu iznešanu no pagraba. Burts norāda ievārījuma veidu un skaitlis - nesamo burciņu skaitu ar mīnuszīmi.

- 3) `?<tukšumsimbols><naturāls skaitlis>`

Apraksta vecmāmiņas jautājumu par noteiktā plaukta vietā esošu burciņu. Skaitlis norāda vietas numuru uz plaukta un nepārsniedz 2147483647.

Ir zināms, ka sākumā plaukts bija tukšs un visas darbības var izpildīt (plaukts nepārpildās un vienmēr ir iznešanai nepieciešamais noteiktā ievārījuma veida burciņu skaits), kā arī tas, ka vecmāmiņa ir uzdevusi vismaz vienu un ne vairāk kā 100 jautājumus.

Izvaddati

Teksta failam `burcinas.rez` jāsatur tieši tik rindas, cik ievaddatos ir "jautājuma" darbību (kas sākas ar '?'). Katram i ($1 \leq i \leq ?$ darbību skaits) faila i -tajā rindā jābūt vienam burtam, kas norāda tā ievārījuma veidu, kas i -tā pēc kārtas jautājuma uzdošanas brīdī atrodas šajā jautājumā norādītajā vietā. Ja vieta ir tukša, i -tajā rindā jāizvada '.' (punkts).

Piemērs (atbilst zīmējumam)

| Ievaddati (fails <code>burcinas.dat</code>) | Izvaddati (fails <code>burcinas.rez</code>) |
|--|--|
| 13 | . |
| A 5 | P |
| ? 237845 | Z |
| P 3 | Z |
| A -2 | A |
| ? 6 | Z |
| Z 4 | . |
| ? 4 | P |
| ? 4 | |
| P -2 | |
| ? 3 | |
| ? 10 | |
| ? 8 | |
| ? 6 | |

2. "SKAITĻU ROTĒŠANA"

Par naturāla skaitļa A rotēšanu saucim tādu operāciju, kuras rezultātā tiek veidots jauns naturāls skaitlis, A pirmo ciparu pārnesot uz skaitļa beigām un atmetot 0 (ja tādas ir) pieraksta sākumā. Tā, rotēšanas rezultātā no skaitļa 3001 iegūst 13, no skaitļa 727 - 277, bet no skaitļa 9 - 9.

Par naturāla skaitļa A rotēšanu k reizes (kur k - naturāls skaitlis) saucim tādu k rotēšanas operāciju virkni $A \rightarrow A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_k$, kur katra nākošā operācija tiek pielietota iepriekšējās operācijas rezultātam.

Uzrakstiet programmu, kas dotam A atrod skaitli A_N , kas iegūts, A rotējot N reizes!

Ievaddati

Teksta faila `rotasana.dat` pirmajā rindā dots naturāls skaitlis A ($1 \leq A \leq 10^{250}$).

Faila otrajā rindā dots naturāls skaitlis N ($1 \leq N \leq 10^{250}$).

Izvaddati

Teksta failā `rotasana.rez` vienīgajā rindā jāizvada naturāls skaitlis, kas iegūts, A rotējot N reizes.

Piemēri

| Ievaddati (fails <code>rotasana.dat</code>) | Izvaddati (fails <code>rotasana.rez</code>) |
|--|--|
| 3001 7 | 13 |

| Ievaddati (fails <code>rotasana.dat</code>) | Izvaddati (fails <code>rotasana.rez</code>) |
|--|--|
| 1203045067089 3 | 45067089123 |

| Ievaddati (fails <code>rotasana.dat</code>) | Izvaddati (fails <code>rotasana.rez</code>) |
|--|--|
| 123456789123456789 123456789123456789 | 123456789123456789 |

3. "VIRKŅU SUMMĒŠANA"

Par divu simbolu virkņu konkatenēšanu jeb *summēšanu* sauc vienas virknes pierakstīšanu otras galā. Šīs darbības apzīmēšanai lieto '+' zīmi. Piemēram, virkņu "abc" un "def" summa ir virkne "abcdef" vai, pierakstot īsāk, "abc"+"def"="abcdef".

Dotas vairākas, savā starpā atšķirīgas, simbolu virknes. Uzrakstiet programmu, kas noskaidro, vai starp šīm virknēm ir atrodamas trīs tādas dažādas virknes A, B un C, ka tām ir spēkā sakarība $A+B=C$! Gadījumā, ja šādas trīs dažādas virknes var izvēlēties vairākos veidos, jāizvada informācija par jebkuru vienu šādu virkņu komplektu.

Ievaddati

Teksta faila `virtnes.dat` pirmajā rindā dots naturāls skaitlis N (virkņu skaits, $3 \leq N \leq 5000$). Nākošajās N rindās dotas virknes (pa vienai virknei katrā rindā), kuras sastāv no latīņu alfabēta mazajiem burtiem. Faila $i+1$ -ajā rindā dota i -tā virkne. Katras virknes garums ir vismaz viens un ne vairāk kā 500 simbolu.

Izvaddati

Teksta faila `virtnes rez` vienīgajā rindā jāizvada vārds "NAV", ja trīs dažādas virknes ar minēto īpašību atrast nav iespējams. Ja trīs dažādas virknes A, B un C ar iepriekšaprakstīto īpašību atrast ir iespējams, tad faila vienīgajā rindā jāizvada trīs naturāli skaitļi: virknes A indekss ievaddatu failā, virknes B indekss ievaddatu failā un virknes C indekss ievaddatu failā. Starp katriem diviem blakus skaitļiem jābūt tieši vienam tukšumsimbolam.

Piemēri

| Ievaddati (fails <code>virtnes.dat</code>) | Izvaddati (fails <code>virtnes rez</code>) | Piezīmes |
|---|---|----------------|
| 5 aab bzx ab a kirils | 4 3 1 | "a"+"ab"="aab" |

| Ievaddati (fails <code>virtnes.dat</code>) | Izvaddati (fails <code>virtnes rez</code>) |
|---|---|
| 4 zbab ab zba abab | NAV |

**LATVIJAS 19. INFORMĀTIKAS OLIMPIĀDES III POSMA
UZDEVUMU APSKATS
Otrā diena (2006.gada 21.marts)**



| Uzdevuma nosaukums: | DISKETES | ARTŪRA SALU VALSTS | HOKEJS |
|--|--|--|--|
| Ievaddatu faila nosaukums: | disketes.dat | asv.dat | hokejs.dat |
| Izvaddatu faila nosaukums: | disketes.rez | asv.rez | hokejs.rez |
| Izpildes laika ierobežojums vienam testam (laiks tiek mērīts uz testēšanas servera): | 1 sekunde | 1 sekunde | 1 sekunde |
| Atmiņas ierobežojums: | 64MB | 64MB | 64MB |
| Maksimāli iespējamais punktu skaits par uzdevumu: | 100 | 100 | 100 |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā PASCAL : | { task: disketes lang: pascal } | { task: asv lang: pascal } | { task: hokejs lang: pascal } |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā C : | /* task: disketes lang: c */ | /* task: asv lang: c */ | /* task: hokejs lang: c */ |
| Komentārs, kas jāiekļauj programmas koda sākumā (pirmajās četrās rindās), ja programma rakstīta valodā C++ : | /* task: disketes lang: c++ */ | /* task: asv lang: c++ */ | /* task: hokejs lang: c++ */ |
| Nosacījums, lai testēšanas serveris atzītu programmu par derīgu testēšanai: | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem | Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajam piemēram |

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili atrodas tekošajā katalogā) un tieši tā, kā norādīts uzdevuma formulējumā (**ar mazajiem burtiem**)!

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL: FreePascal (versija 1.0.10) ar parametriem -So -Sg -O2 -Xs

Valodai C/C++: GNU C/C++ (versija 3.3.6) ar parametriem -O2 -static -lm

Jautājumus par uzdevumu formulējumiem centieties uzdot sacensību pirmās stundas laikā!

Katra uzdevuma labāko programmas kodu pēc nosūtīšanas uz serveri saglabājiēt arī darba datorā un neizdzēsiet pēc sacensību beigām!

LATVIJAS 19. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Otrā diena (2006.gada 21.marts)



1. "DISKETES"

Līdz ar rajona informātikas olimpiādes uzdevumu tekstiem olimpiādes žūrija nosūta arī disketes ar elektroniskajiem materiāliem. Tā kā disketes mēdz nelasīties, šogad nolemts katram rajonam nosūtīt K disketes ar identisku saturu. Žūrijas rīcībā ir dažāds skaits M dažādu ražotāju diskešu.

Diskešu komplektu saucim par *labu*, ja tajā nav divu diskešu no vienas ražotājfirmas.

Uzrakstiet programmu, kas nosaka, kādu lielāko labu diskešu komplektu skaitu iespējams izveidot no dotajām disketēm!

Ievaddati

Teksta faila `disketes.dat` pirmajā rindā doti divi naturāli skaitļi K ($1 \leq K \leq 10^5$) un M ($1 \leq M \leq 10^5$), kas atdalīti ar tukšumsimbolu.

Katrā no nākošajām M faila rindām dots vienas ražotājfirmas diskešu skaits. i -tās ražotājfirmas ($1 \leq i \leq M$) diskešu skaits s_i ($1 \leq s_i \leq 10^4$) dots faila $i+1$ -ajā rindā.

Izvaddati

Teksta faila `disketes.rez` vienīgajā rindā jāizvada vesels nenegatīvs skaitlis - lielākais labo diskešu komplektu skaits, ko iespējams izveidot no dotajām disketēm.

Piemēri

| Ievaddati (fails <code>disketes.dat</code>) | Izvaddati (fails <code>disketes.rez</code>) |
|--|--|
| 2 3 6 8 4 | 9 |

| Ievaddati (fails <code>disketes.dat</code>) | Izvaddati (fails <code>disketes.rez</code>) |
|--|--|
| 3 1 10000 | 0 |

| Ievaddati (fails <code>disketes.dat</code>) | Izvaddati (fails <code>disketes.rez</code>) |
|--|--|
| 2 2 10000 13 | 13 |

2. "ARTŪRA SALU VALSTS"

Artūra Salu Valsts (ASV) teritoriju veido N sīku salu arhipelāgs. Karalis Artūrs ir nolēmis valstī izveidot modernu ceļu infrastruktūru un tāpēc nepieciešams starp salām uzbūvēt tiltus. Ir zināms, ka sākumā neviena tilta nav. Lai ierēdņiem atvieglotu lēmumu pieņemšanu, karalis ir izdevis likumu, ka:

- 1) starp katrām divām salām drīkst uzbūvēt ne vairāk kā vienu tiltu;
- 2) katrā brīdī drīkst būt tikai viens – īsāko – no vēl neuzbūvētajiem tiltiem, taču, ja šādi tilti ir vairāki, tad drīkst izvēlēties jebkuru no tiem;
- 3) būvniecība jāpārtrauc, kad visas salas kļūst savienotas savā starpā (no katras salas kļūst iespējams aizbraukt uz jebkuru citu salu, pārvietojoties tikai pa tiltiem).

Katru salu var uzskatīt par punktvēda objektu, kura koordinātas ir veseli skaitļi, un tilta garums ir attālums starp tā galapunktiem.

Uzrakstiet programmu, kas nosaka, kāds mazākais tiltu skaits jāuzceļ, lai būtu izpildīti visi likuma nosacījumi!

Ievaddati

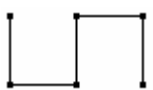

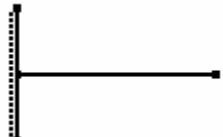
Teksta faila `asv.dat` pirmajā rindā dots naturāls skaitlis N (salu skaits, $1 < N \leq 5000$).

Katrā no nākošajām N rindām dotas vienas salas koordinātas - divi veseli skaitļi X_i un Y_i , kas atdalīti ar tukšumsimbolu. Faila $i+1$ -ajā rindā ($1 \leq i \leq N$) dotas i -tās salas koordinātas. Zināms, ka nav divu salu ar vienādām koordinātām un ka neviena koordināta pēc absolūtās vērtības nepārsniedz 15000.

Izvaddati

Teksta faila `asv.rez` vienīgajā rindā jāizvada naturāls skaitlis - mazākais iespējamais tiltu skaits.

Piemēri

| Ievaddati (fails <code>asv.dat</code>) | Izvaddati (fails <code>asv.rez</code>) | Piezīmes |
|--|---|---|
| 6 1 1 1 2 2 1 2 2 3 1 3 2 | 5 | Visi tilti garumā 1. Viens no variantiem:  |
| 10 -1000 73 -1000 74 1000 73 1000 74 2999 73 2999 74 -1001 73 -1001 74 999 73 999 74 | 15 | 9 tilti garumā 1, 4 tilti garumā $\sqrt{2}$, 2 tilti garumā 1999. Viens no variantiem:  |
| 4 0 0 3 0 0 1 0 -1 | 4 | 2 tilti garumā 1, 1 tilts garumā 2, 1 tilts garumā 3.  |

3. "HOKEJS"

Pensionētā skolotāja Alma ir liela hokeja cienītāja. Pat tad, kad Alma ciemojas Lietuvā, viņa ar nepacietību gaida katru ziņu par Latvijas hokeja izlases startu. Tā kā kaimiņvalstī ne visur ir iespējams uztvert Latvijas TV un radio raidījumus, tad skolotāja izmanto Tīmekli. Alma ir ievērojusi, ka visātrāk pašreizējo spēles rezultātu iespējams uzzināt viņas iecienītajā ziņu portālā, pie kam nevis sporta ziņās, bet lasītāju atsauksmju sadaļā.

Piemēram, jau pāris sekundes pēc iesistiem vārtiem var parādīties ziņa:

"Mūsējie rullē - 1:0 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!".

Diemžēl, atsauksmes mēdz rakstīt ne tikai nopietni cilvēki, bet arī jokdari, kuri raksta nepatiesības. Piemēram, tūlīt pēc ziņas par 1:0 var parādīties ziņa:

"Pilnīgs ārprāts - jau kāšam ar 0:2!!!!".

Tā kā hokejā vārtus var tikai iesist, bet nevar "paņemt atpakaļ", tad spēlē rezultātam $a_1:b_1$ rezultāts $a_2:b_2$ var hronoloģiski sekot tikai tad, ja $a_2 \geq a_1$ un $b_2 \geq b_1$ (a_1, a_2, b_1, b_2 – veseli nenegatīvi skaitļi).

Savācot hronoloģiskā secībā visus atsauksmēs minētos rezultātus, var veidoties, piemēram, šāda pretrunīga aina: 1:0, 0:2, 4:3, 1:0, 1:1, 1:2, 0:3, 3:3, 2:3, 2:4, 4:3.

Visi atsauksmju rakstītāji ir vienas komandas līdzjutēji, tāpēc savējo iemesto vārtu skaits vienmēr tiek rakstīts pirms kolona, bet pretinieka - aiz.

Protams, ka skolotāja Alma varētu sagaidīt autoritatīvu ziņu sava iecienītā portāla sporta sadaļā, bet tam viņai trūkst pacietības. Viņa vadās pēc savas pārlicības, ka īstais gala rezultāts komentāros parādīsies. Skolotāja par *ticamu galarezultātu* ir nolēmusi uzskatīt to rezultātu, kurš ir parādījies komentāros un kuram hronoloģiski atbilst visvairāk rezultātu. Piemēram, augstāk dotajā piemērā vienlīdz ticami rezultāti ir 2:4 un 4:3, jo katram no tiem var atrast sešus hronoloģiski nepretrunīgus rezultātus: 1:0 → 1:0 → 1:1 → 1:2 → 2:3 → 2:4 vai 1:0 → 1:0 → 1:1 → 1:2 → 3:3 → 4:3.

Uzrakstiet programmu, kas nosaka, kāds (vai kādi) ir ticamie spēles galarezultāti!

Ievaddati

Teksta faila `hokejs.dat` pirmajā rindā dots naturāls skaitlis R (rezultātu skaits, $R \leq 10000$). Katrā no nākošajām R faila rindām dots viens komentāros sastaptais spēles rezultāts formā `<vesels_nenegatīvs_skaitlis>:<vesels_nenegatīvs_skaitlis>`, kur pirmais skaitlis apzīmē savējo komandas gūto vārtu skaitu, bet otrais - pretinieku komandas gūto vārtu skaitu. Gūto vārtu skaits visos rezultātos ir robežās no 0 līdz 99. Visi rezultāti ir doti hronoloģiskā secībā tā kā tie parādījušies komentāros, sākot ar agrāko.

Izvaddati

Teksta failā `hokejs.rez` jāizvada ticamie galarezultāti tādā secībā, kādā tie pirmoreiz parādījušies komentāros. Katrs rezultāts jāizvada savā rindā tādā pat formātā, kāds rezultātu ievadei tika izmantots ievaddatos.

Piemērs

| Ievaddati (fails <code>hokejs.dat</code>) | Izvaddati (fails <code>hokejs.rez</code>) |
|--|--|
| 11 | 4:3 |
| 1:0 | 2:4 |
| 0:2 | |
| 4:3 | |
| 1:0 | |
| 1:1 | |
| 1:2 | |
| 0:3 | |
| 3:3 | |
| 2:3 | |
| 2:4 | |
| 4:3 | |