GeOlymp

Series 2013 Episode III

#	Problem Name	Time Limit	Memory Limit
1	specialist	1 sec.	64 MB
2	astrapark	1 sec.	64 MB
3	binary	1 sec.	64 MB
4	roads	1 sec.	64 MB
5	board	1 sec.	64 MB

ამოცანა A. "სპეციალისტი"

უნიჭო გიორგი ფუნდამენტური პროგრამირების მსოფლიო მაშტაბის სპეციალისტია. ამ სტატუსის შესანარჩუნებლად მას ზოგჯერ უწევს საოლიმპიადო ამოცანების შედგენა ზღვისიქითური უაღრესად მნიშვნელოვანი შეჯიბრებებისთვის.

ერთ-ერთი შეჯიბრი უკვე ახლოვდება. უნიჭო გიორგის სჭირდება N ცალი საოლიმპიადო ამოცანა ამ შეჯიბრზე წარმოსადგენად. მან უკვე ამოწერა ამოცანების იდეები სავარჯიშოების ძველი წიგნებიდან და მიმართა თავის M რაოდენობის მოსწავლეს. ყოველ მათგანს გიორგიმ თავაზიანად სთხოვა რამდენიმე ამოცანის ამომხსნელი პროგრამული კოდის და ტესტების კრებულის შედგენა. კერძოდ, i-ურ ($1 \le i \le M$) მოსწავლეს მან X_i ცალი ამოცანის მომზადება სთხოვა. ყველა მოსწავლემ სხვადასხვა ამოცანები მიიღო გასამზადებლად. შევნიშნოთ, რომ ჯამში გიორგიმ შეიძლება N-ზე მეტი ამოცანაც მისცა თავის მოსწავლეებს, ბოლო-ბოლო გავარჯიშება არ აწყენთ.

სამწუხაროდ, მანაოს ალერგია აქვს ასეთი მაღალი დონის სპეციალისტებზე. ეს ცნობილია უნიჭო გიორგისთვისაც და შიშობს, რომ მისი ზოგიერთი მოსწავლე შეიძლება მანაოს ზეგავლენის ქვეშ მოექცეს და ამოცანები აღარ მოუმზადოს. ცხადია, თუ გიორგის N რაოდენობის ამოცანა არ დაუგროვდება, იგი შეჯიბრს ვეღარ მოაწყობს. ამრიგად მას აინტერესებს, სულ მცირე რამდენი მოსწავლე უნდა მოექცეს მანაოს ზეგავლენის ქვეშ, რომ შეჯიბრის ჩატარებას საფრთხე შეექმნას. სხვა სიტყვებით რომ ვთქვათ, საჭიროა ისეთი მინიმალური X რიცხვის პოვნა, რომ მოსწავლეების ერთი მაინც X ზომის ქვესიმრავლე არსებობდეს, რომლის მანაოს ზეგავლენის ქვეშ მოქცევისას უნიჭო გიორგის აღარ დაუგროვდება N რაოდენობის ამოცანა. დაეხმარეთ მას ამ ამოცანის გადაჭრაში.

შეზღუდვები

$1 \le N, M \le 100$

$1 \le X_i \le 100$ ყოველი i-სთვის 1.. M შუალედში. ყველა X_i-ს ჯამი N-ზე მეტი ან ტოლია.

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების specialist.in ფაილის პირველ ხაზში წერია ერთი ჰარით გამოყოფილი ორი მთელი რიცხვი N და M. ფაილის მეორე ხაზში წერია თითო ჰარით გამოყოფილი M რაოდენობის რიცხვი - X_i მიმდევრობა. გარანტირებულია, რომ ეს მონაცემები ამოცანის შეზღუდვებს აკმაყოფილებენ.

გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების specialist.out ფაილში დაბეჭდეთ ზეგავლენის ქვეშ მოქცეული მოსწავლეების მინიმალური რაოდენობა, რომლის გამოც უნიჭო გიორგის შეიძლება N-ზე ნაკლები ამოცანა მოუმზადონ.

შემომავალი ფაილის მაგალითი (specialist.in)	გამომავალი ფაილის მაგალითი (specialist.out)
8 4 4 5 2 4	2

განმარტება.

ნებისმიერი ერთი მოსწავლე რომ ზეგავლენის ქვეშ მოექცეს, უნიჭო გიორგის 8 ამოცანას მაინც მოუმზადებენ. მაგრამ მაგალითად პირველი ორი მოსწავლე თუ აღარ მოამზადებს ამოცანებს, უნიჭო გიორგის მხოლოდ 6 ცალი დაუგროვდება.

ამოცანა B. “ასტრა პარკი”

ლევანის და თორნიკეს ძალიან უყვართ ასტრა პარკში კარტინგებზე სიარული. კარტინგი არის მინი-რბოლის ანალოგი: რამდენიმე ადამიანი ეჯიბრება ერთმანეთს პატარა სარბოლო მანქანებით. ტრასა წრიული ფორმისაა და ყოველი მრბოლელი ცდილობს, რაც შეიძლება მეტი წრე დაასრულოს მოცემულ დროში. ანუ წრის დამთავრების შემდეგ მრბოლელი არ ჩერდება და აგრძელებს მოძრაობას შემდეგ წრეზე. როდესაც დრო მთავრება, მრბოლელებს აჩერებენ და წრეს აღარ ამთავრებინებენ. რბოლისას ავტომატურად აღირიცხება, თუ რა დროში გაიარა ყოველმა მათგანმა ყოველი წრე.

ლევანმა და თორნიკემ ახლახანს დაამთავრეს რბოლა და მიიღეს სტატისტიკის ფურცელი, სადაც წერია ვინ რომელი წრის შემოვლას რამდენი ხანი მოანდომა წამებში. ისინი დაინტერესდნენ, ჯამური დისტანციის თვალსაზრისით სულ მცირე რამდენჯერ მოახდინა მეთოქის გადასწრება თორნიკემ და რამდენჯერ ლევანმა. მაგალითად, თუ ლევანმა ორი წრე გაიარა შესაბამისად 25 და 22 წამში, ხოლო თორნიკემ პირველი წრე 20 წამში დაამთავრა, ხოლო მეორე 30 წამში, ჩვენ შეგვიძლია დავასკვნათ, რომ თავიდან თორნიკე წინ იყო, ხოლო მეორე წრეზე რაღაც მომენტში ლევანმა გაუსწრო. ამრიგად, ბიჭებმა სულ მცირე თითოჯერ მოახერხეს ერთმანეთისთვის გასწრება.

შევნიშნოთ, რომ დაწევა გასწრებად არ ითვლება. ასევე გასწრებად არ ითვლება, თუ ერთ-ერთი ყმაწვილი წინ იყო, შემდეგ მეორე მას დაეწია (მაგრამ არ გადაუსწრო) და შემდეგ პირველი ისევ წინ გავიდა. დააკვირდით, რომ გასწრებას ჯამური მანძილით ვითვლით და არა ფიზიკურად მანქანების ტრასაზე შეხვედრით.

თქვენ მოცემული გაქვთ თორნიკეს და ლევანის მიერ ყოველი წრის გავლის დრო წამებში ქრონოლოგიური მიმდევრობით. თორნიკემ N რაოდენობის წრე დაამთავრა, ხოლო ლევანმა კი M რაოდენობის. გამოთვალეთ, მინიმუმ რამდენჯერ გავიდა წინ თორნიკე და რამდენჯერ - ლევანი.

შეზღუდვები

$1 \leq N, M \leq 100$

ლევანის და თორნიკეს მიერ წრეების გავლის დროები მთელი რიცხვებია [15, 100] შუალედში.

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების astrapark.in ფაილის პირველ სტრიქონში ჩაწერილია ერთი ჰარით გამოყოფილი ორი მთელი N და M რიცხვი. მეორე სტრიქონში წერია თითო ჰარით გამოყოფილი N რაოდენობის რიცხვი - ლევანის მიერ წრეების გავლის დროები ქრონოლოგიური მიმდევრობით. მესამე სტრიქონში იგივე ფორმატში ჩაწერილია M რაოდენობის რიცხვი -

თორნიკეს მიერ წრეების გავლის დროები ქრონოლოგიური მიმდევრობით. გარანტირებულია, რომ ეს მონაცემები ამოცანის შეზღუდვებს აკმაყოფილებენ.

გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა astrapark.out ფაილში დაბეჭდეთ ერთი ჰარით გამოყოფილი ორი მთელი რიცხვი. მათგან პირველი უნდა იყოს გასწრებების მინიმალური შესაძლო რაოდენობა, რომლებიც მოახდინა ლევანმა, ხოლო მეორე - იგივე სიდიდე თორნიკესთვის.

შემომავალი ფაილის მაგალითი (astrapark.in)	გამომავალი ფაილის მაგალითი (astrapark.out)
4 4 20 15 30 30 20 25 20 31	1 0
3 4 15 17 15 16 15 18 15	2 2
1 4 100 15 15 15 15	0 1

განმარტება.

პირველ მაგალითში ლევანმა პირველი წრე 20 წამში, შემდეგი 15 წამში, შემდეგი ორი კი თითო 30 წამში გაიარა. თორნიკემ წრეებს შესაბამისად 20, 25, 20 და 31 წამი მოანდომა. აქედან შეგვიძლია დავასკვნათ, რომ რბოლა მინიმუმ 96 წამს გრძელდებოდა. ვინაიდან ბიჭებმა პირველი წრე ერთსადაიმავე დროში დაამთავრეს, შესაძლებელია რომ მათ მთელი გზა ისე გაიარეს, რომ ვერც ერთმა ვერ გაუსწრო მეორეს. მეორე წრეზე აშკარად ლევანი გავიდა წინ. მესამე წრე ორივე მრბოლელმა ერთსადაიმავე დროს დაამთავრა, ხოლო შემდეგ ისევ ლევანი აღმოჩნდა უფრო სწრაფი. ვინაიდან საუკეთესო შემთხვევაში თორნიკე ლევანის წინ არ ყოფილა, ლევანს მხოლოდ პირველი გასწრება ჩაეთვალა და ანგარიშია 1:0.

მეორე მაგალითში თავიდან ლევანი იყო წინ, შემდეგ თორნიკემ გადაუსწრო, შემდეგ ისევ ლევანი დაწინაურდა და ბოლოს თორნიკემ კვლავ გადაუსწრო. შეიძლება იქ მეტი გასწრებაც მომხდარიყო, მაგრამ ჩვენ ამის შესახებ მოცემული ინფორმაციიდან ვერ დავასკვნით.

მესამე მაგალითში ჩვენ დანამდვილებით შეგვიძლია მხოლოდ იმის თქმა, რომ თორნიკემ ერთხელ გაუსწრო ლევანს. დააკვირდით, რომ თორნიკემ მეხუთე წრის გავლა 40 წამშიც ვერ მოასწრო.

ამოცანა C. “მანაო და ორობითი რიცხვები”

მანაო ხშირად თამაშობს ორობით რიცხვებთან. ამჯერად მან ასეთი თამაში გამოიგონა: იგი ირჩევს ორ მთელ დადებით რიცხვს X და Y. შემდეგ იგი იღებს ცარიელ S სტრიქონს და ასრულებს შემდეგ პროცედურას ორჯერ ან მეტჯერ:

1) T := binary(X), ანუ T სტრიქონში ინახავს X რიცხვის ორობითი თვლის სისტემაში წარმოდგენას.
2) S := S || T, ანუ S სტრიქონს მარჯვნიდან უმატებს T სტრიქონს.
3) X := X + Y, ანუ X რიცხვის როლში აწი (X+Y) რიცხვი იმოქმედებს.

ახლა კი მოდით შებრუნებული ამოცანა ამოვხსნათ. მოცემული *S* ორობითი სტრიქონისთვის აღვადგინოთ ის (X, Y) წყვილები, რომლების მეშვეობით მანაოს თავისი თამაშით შეიძლება ეს *S* სტრიქონი მიეღო. ჩვენ გვაინტერსებს ასეთი წყვილების საერთო რაოდენობა.

შეზღუდვები

S სტრიქონის სიგრძე არ აღემატება 60 სიმბოლოს.

S სტრიქონი მხოლოდ ‘0’ და ‘1’ სიმბოლოებისგან შედგება და იწყება ‘1’-ით.

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების `binary.in` ფაილის პირველ ხაზში ჩაწერილია *S* სტრიქონი. გარანტირებულია, რომ იგი აკმაყოფილებს ამოცანის შეზღუდვებს.

გამომავალი ფაილის ფორმატი

გამოსატან მონაცემთა `binary.out` ფაილში ჩაწერეთ ისეთი (X, Y) წყვილების რაოდენობა, რომლებისთვის მანაოს თამაშის შედეგად შეიძლება *S* სტრიქონი მიეღო.

შემომავალი ფაილის მაგალითი (binary.in)	გამომავალი ფაილის მაგალითი (binary.out)
10110011101	3
1010	0

განმარტება.

პირველი მაგალითი. მანაოს თამაშის შედეგად “10110011101” სტრიქონი მიიღება შემდეგი სამი (X, Y) წყვილისგან:

(2, 411). ორობითი თვლის სისტემაში 2 არის “10“, ხოლო 2+411=413 არის „110011101“. ესე იგი, ამ წყვილისთვის პროცედურის ორჯერ შესრულებით მიიღება მოცემული სტრიქონი.

(5, 4). 5 = “101”, 5+4 = “1001”, 5+4+4 = “1101”.

(5, 152). 5 = “101”, 5+152 = “10011101”.

ამოცანა D. “გზები”

ერთ უცნაურ ქვეყანაში არის N რაოდენობის ქალაქი. ამ ქალაქების ზოგიერთი წყვილი შეერთებულია ორმხრივი გზით. ქალაქები დანომრილია 1-დან N-მდე მთელი რიცხვების მეშვეობით. ყოველი ქალაქის შემოსასვლელზე წერია ერთიდაიგივე შინაარსის ფრაზა, კერძოდ i-ური ქალაქის შემოსასვლელს აწერია „ამ ქალაქიდან თქვენ ვერ მოხვდებით A_i ქალაქში”. ეს წარწერა გულისხმობს, რომ i-ური ქალაქიდან A_i-ურ ქალაქში არ არსებობს არც პირდაპირი გზა, არც გზების მიმდევრობა, რომელიც ამ ორ ქალაქს აკავშირებს. A_i მიმდევრობა განსხვავებული ნომრებისგან შედგება.

თქვენი ამოცანაა, მოცემულ ინფორმაციაზე დაყრდნობით დაადგინოთ, რა მაქსიმალური რაოდენობის გზა შეიძლება არსებობდეს ამ ქვეყანაში.

შეზღუდვები

$2 \leq N \leq 100$

$1 \leq A_i \leq N$, $A_i \neq$ i ყოველი i-სთვის 1.. N შუალედში.

$A_i \neq A_j$ ყოველი (i, j) წყვილისთვის, სადაც $1 \leq$ i < j $\leq N$.

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების roads.in ფაილის პირველ ხაზში წერია ერთი მთელი რიცხვი N. ფაილის მეორე ხაზში წერია თითო ჰარით გამოყოფილი N რაოდენობის რიცხვი - A_i მიმდევრობა. გარანტირებულია, რომ ეს მონაცემები ამოცანის შეზღუდვებს აკმაყოფილებენ.

გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების roads.out ფაილში დაბეჭდეთ გზების მაქსიმალური რაოდენობა, რომლებიც შეიძლება იყოს ქვეყანაში.

შემომავალი ფაილის მაგალითი (roads.in)	გამომავალი ფაილის მაგალითი (roads.out)
4 2 1 4 3	2

განმარტება.

ქვეყანაში შეიძლება იყოს გზები ქალაქების შემდეგ წყვილებს შორის: (1, 3), (2, 4). კიდევ ერთი შესაძლო ვარიანტი 2 გზით არის (1, 4), (2, 3). მეტი გზა ქვეყანაში ვერ იქნება, ვინაიდან დაირღვევა წარწერებით განსაზღვრული პირობები.

ამოცანა E. “შავ–თეთრი დაფა”

მოცემულია უჯრედებიანი დაფა, რომელიც შედგება $2N$ სტრიქონისა და $2M$ სვეტისაგან. სტრიქონები და სვეტები გადანომრილია ზევიდან ქვევით და მარცხნიდან მარჯვნივ, შესაბამისად 1–დან $2N$–მდე და 1–დან $2M$-მდე მთელი რიცხვებით. ის უჯრები, რომლის სვეტისა და სტრიქონის ნომრების ჯამი ლუწია, შეღებილია შავად, ხოლო ყველა სხვა კი თეთრად. ასევე დაფის თითოეულ უჯრაში წერია 1–დან 10–მდე რიცხვი.

თქვენი ამოცანაა, დაფა ისე გარდაქმნათ, რომ შავ უჯრებში მდებარე რიცხვების ჯამი რაც შეიძლება მეტი გახდეს. ამისთვის შეგიძლიათ შეასრულოთ შემდეგი ორი ტიპის ოპერაციები:

1) ნებისმიერ სტრიქონში ყოველი რიცხვი გადაანაცვლოთ მის მარჯვნივ მდებარე უჯრაზე, ხოლო ამ სტრიქონის ბოლო რიცხვი ჩაწეროთ ამავე სტრიქონის პირველ უჯრაში.

2) ნებისმიერ სვეტში ყოველი რიცხვი გადაანაცვლოთ მის ქვევით მდებარე უჯრაზე, ხოლო ამ სვეტის ბოლო რიცხვი ჩაწეროთ სვეტის პირველ უჯრაში.

თქვენ მოცემული გაქვთ საწყისი დაფა. იპოვეთ მასზე ჩატარებული დაშვებული ოპერაციების შედეგად შავ უჯრებში ჩაწერილი რიცხვების რა მაქსიმალურ ჯამს მიიღებთ. დაბეჭდეთ შესაბამისი ოპერაციების მიმდევრობა. თქვენი მიმდევრობა უნდა შეიცავდეს არაუმეტეს 100,000 ოპერაციას.

შეზღუდვები

$1 \leq N, M \leq 5$

დაფის უჯრებში მდებარე ყოველი რიცხვი მთელია და იმყოფება 1..10 შუალედში.

გარანტირებულია, რომ წინა შეზღუდვების შესაბამისი ყველა დაფისთვის არსებობს სვლების მიმდევრობა, რომელიც ახდენს შავ უჯრებში მდებარე რიცხვების ჯამის მაქსიმიზაციას და შეიცავს 100,000 ან ნაკლებ ოპერაციას.

შემომავალი ფაილის ფორმატი

შესატანი მონაცემების `board.in` ფაილის პირველ ხაზში წერია ერთი ჰარით გამოყოფილი ორი მთელი რიცხვი N და M. შემდეგი $2N$ ხაზიდან თითოეულში წერია $2M$ ცალი ერთმანეთისგან თითო ჰარით გამოყოფილი მთელი რიცხვი. i-ური ხაზის j–ური რიცხვი შეესაბამება დაფის i-ური სტრიქონსა და j-ური სვეტის თანაკვეთაზე მდებარე რიცხვს. გარანტირებულია, რომ ეს მონაცემები ამოცანის შეზღუდვებს აკმაყოფილებენ.

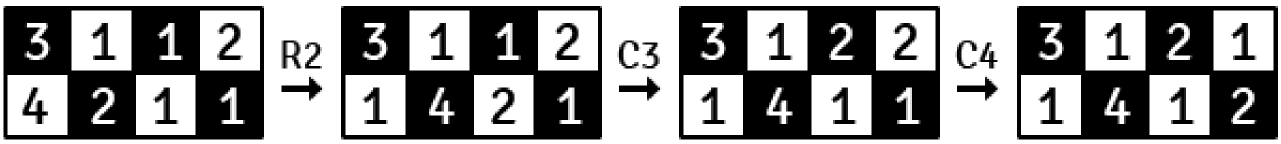
გამომავალი ფაილის ფორმატი

გამოსატანი მონაცემების `board.out` ფაილში უნდა დაბეჭდეთ სვლების რაოდენობა K. შემდეგი K ხაზიდან თითოეულში დაბეჭდეთ ერთი სვლა: R<X> ან C<Y>. R<X> ნიშნავს ოპერაციის ჩატარებას სტრიქონზე ნომრით <X>, ხოლო C<Y> ოპერაციის ჩატარებას სვეტზე ნომრით <Y>. K

არ უნდა აღემატებოდეს 100,000–ს. სვლები უნდა იყოს დაბეჭდილი ზუსტად იმ მიმდევრობით, რა მიმდევრობითაც საჭიროა მათი შესრულება. რამოდენიმე შესაძლო მიმდევრობის არსებობის შემთხვევაში შეგიძლიათ ნებისმიერი გამოიტანოთ.

შემომავალი ფაილის მაგალითი (board.in)	გამომავალი ფაილის მაგალითი (board.out)
1 2 3 1 1 2 4 2 1 1	3 R2 C3 C4

განმარტება.



შავ უჯრებში მდებარე რიცხვების ჯამია 3 + 2 + 4 + 2 = 13. ამ დაფაზე დაშვებული ოპერაციების შედეგად უკეთესი ჯამის მიღება შეუძლებელია.