

Magic City

ტაშკენტის მერს ცნობილი გასართობი პარკის, Magic City-ს ხელახალი დაგეგმარება სურს. მოცემული გაქვთ დადებითი მთელი რიცხვი K , თქვენი ამოცანაა დააპროექტოთ პარკი შემდეგნაირად:

- აირჩიეთ მთელი რიცხვი N , ატრაქციონების რაოდენობა. ატრაქციონები გადანომრილია მთელი რიცხვებით 0-დან $N - 1$ -ის ჩათვლით.
- დაამატეთ ორმხრივი ბილიკები **განსხვავებული** ატრაქციონების წყვილებს შორის. **სავალდებულო არაა, რომ ატრაქციონების ყველა წყვილი ერთმანეთისგან მიღწევადი იყოს საფეხმავლო ბილიკების გამოყენებით.**
- თითოეული i ისთვის, რომლისთვისაც $0 \leq i < N$, i -ურ ატრაქციონს მიანიჭეთ $T[i]$ ტიპი. არსებობს $2K$ ცალი ტიპი, რომლებიც დანომრილია 0-დან $2K - 1$ -ის ჩათვლით. თითოეული ტიპი უნდა მიენიჭოს მინიმუმ ერთ ატრაქციონს. სხვადასხვა ატრაქციონს შეიძლება ჰქონდეს ერთი და იგივე ტიპი.

ბაზრის კვლევამ აჩვენა, რომ:

1. თითოეულ ვიზიტორს ურჩევნია სამი ატრაქციონის მონახულება, ერთი და იმავე ტიპის ორი ატრაქციონის ზედიზედ მონახულების გარეშე.
2. ვიზიტორებს არ მოსწონთ ატრაქციონები, რომლებიც ბევრი საფეხმავლო ბილიკის ბოლოა.

ყველა პოტენციური ვიზიტორის დასაკმაყოფილებლად, მერი თქვენს დიზაინს ორ მოთხოვნას უყენებს.

$0 \leq t_1, t_2, t_3 < 2K$ -სთვის, ჩვენ ვუნოდებთ **ტიპების** (t_1, t_2, t_3) დალაგებულ სამეულს **საინტერესოს**, თუ $t_1 \neq t_2$ და $t_2 \neq t_3$. გაითვალისწინეთ, რომ t_1 შეიძლება ტოლი იყოს t_3 -ის. ამგვარად, არსებობს $2K \cdot (2K - 1)^2$ საინტერესო სამეული.

მოთხოვნა 1: თითოეული საინტერესო (t_1, t_2, t_3) სამეულისთვის, უნდა არსებობდეს სამი ატრაქციონი a_1, a_2, a_3 (სადაც, $0 \leq a_1, a_2, a_3 < N$), რომლებისთვისაც:

- a_1, a_2, a_3 -ის ტიპები ემთხვევა t_1, t_2, t_3 -ს. ანუ, $T[a_1] = t_1$, $T[a_2] = t_2$ და $T[a_3] = t_3$.
- a_1 და a_2 ს შორის არის საფეხმავლო ბილიკი.
- a_2 და a_3 ს შორის არის საფეხმავლო ბილიკი.

არ აქვს მნიშვნელობა, არის თუ არა a_1 და a_3 ს შორის საფეხმავლო ბილიკი. ასევე გაითვალისწინეთ, რომ, როდესაც $t_1 = t_3$, a_1 და a_3 ატრაქციონები შეიძლება ერთი და იგივე იყოს.

მოთხოვნა 2: თითოეული ატრაქციონი შეიძლება იყოს **არაუმეტეს K ცალი** საფეხმავლო ბილიკის ბოლო.

ეს ამოცანა შედგება 50 **output only** ქვეამოცანისგან **ნაწილობრივი ქულებით**. თითოეული ქვეამოცანა შეესაბამება K -ს კონკრეტულ მნიშვნელობას და თქვენ უნდა შექმნათ პარკი, რომელიც აკმაყოფილებს ზემოთ ჩამოთვლილ ყველა პირობას K -ს ამ მნიშვნელობისთვის. თქვენი ქულა დამოკიდებულია თქვენს ამოხსნაში არსებული ატრაქციონების რაოდენობაზე - ნაკლები ატრაქციონი უფრო მაღალ ან თანაბარ ქულას იძლევა.

იმპლემენტაციის დეტალები

არსებობს ორი მეთოდი თქვენი ამოხსნის ასატვირთად, და თქვენ შეგიძლიათ აირჩიოთ ნებისმიერი თითოეული ქვეამოცანისთვის:

- პროცედურის გამოძახება
- გამომავალი ფაილი

იმისთვის, რომ თქვენი ამოხსნა ატვირთოთ **პროცედურის გამოძახებით**, უნდა დააიმპლემენტიროთ შემდეგი პროცედურა:

```
std::pair<std::vector<int>, std::vector<std::pair<int, int>>>>  
construct(int K)
```

- K : ატრაქციონების ტიპების რაოდენობის ნახევარი და ასევე თითოეული ატრაქციონისთვის გამომავალი ბილიკების მაქსიმალური რაოდენობა.
- ეს პროცედურა გამოიძახება ზუსტად ერთხელ თითოეული ქვეამოცანისთვის.

პროცედურამ უნდა დააბრუნოს წყვილი (T, E) რომელიც აღწერს გასართობ პარკს. დავუშვათ, M არის თქვენს პარკში საფეხმავლო ბილიკების რაოდენობა.

- T : N სიგრძის მასივი, რომელიც აღწერს ატრაქციონების ტიპებს.
- E : M სიგრძის მასივი, რომელიც აღწერს საფეხმავლო ბილიკებს. თითოეული $0 \leq j < M$ -ისთვის, $E[j] = (U[j], V[j])$ წარმოადგენს ორმხრივ ბილიკს განსხვავებულ $U[j]$ და $V[j]$ ატრაქციონებს შორის. ატრაქციონების ერთსა და იმავე წყვილს შორის დაშვებულია ერთზე მეტი ბილიკის ქონა.

თქვენი ამოხსნის **გამომავალი ფაილის** მეშვეობით ასატვირთად, შექმენით და ატვირთეთ ტექსტური ფაილი შემდეგი ფორმატით:

```

N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]

```

გაითვალისწინეთ, რომ თქვენი ამოხსნა უნდა აკმაყოფილებდეს შემდეგ შეზღუდვებს, რათა **ვალიდურად** ჩაითვალოს:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ თითოეული $0 \leq i < N$ -ისთვის და ყველა ტიპი უნდა მიენიჭოს მინიმუმ ერთ ატრაქციონს.
- $0 \leq U[j], V[j] < N$ და $U[j] \neq V[j]$ თითოეული $0 \leq j < M$.
- მოთხოვნები 1 და 2 უნდა დაკმაყოფილდეს.

შეზღუდვები

- $1 \leq K \leq 50$

შეფასება

ამოცანას აქვს 50 ქვეამოცანა, ერთი თითოეული მთელი K რიცხვისთვის 1-დან 50-ის ჩათვლით. i -ური ქვეამოცანისთვის ($1 \leq i \leq 50$), K -ს მნიშვნელობა არის i .

ქვემოთ მოცემული ცხრილის მიხედვით, თითოეულ ქვეამოცანას აქვს ქულა S და ატრაქციონების სამიზნე რაოდენობა P .

ქვეამოცანა	S	P
1	1	2
2	8	12
3	9	24
4	9	40
5	9	50
6 - 10	4	$12 \cdot K$
11 - 12	3	$12 \cdot K$
13 - 50	1	$12 \cdot K$

თითოეული ქვეამოცანისთვის, თუ თქვენი ამოხსნა არ აღწერს ვალიდურ გასართობ პარკს, მაშინ თქვენი ამოხსნის ქულა იქნება 0 (CMS-ში მოხსენებული იქნება, როგორც Output isn't correct).

სხვა შემთხვევაში, თქვენი ქულა გამოითვლება N -ისა და S და P პარამეტრების მიხედვით შემდეგნაირად:

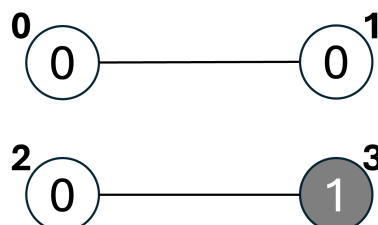
პირობა	ქულა
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

მაგალითი

განვიხილოთ შემდეგი გამოცხება:

```
construct (1)
```

ამ მაგალითში, $K = 1$, ამიტომ არსებობს $2K = 2$ ტიპის ატრაქციონი. ქვემოთ მოცემული სურათი გვიჩვენებს სწორ ამონახსნს $N = 4$ ატრაქციონით და $M = 2$ საფეხმავლო ბილიკებით. ატრაქციონები 0, 1, 2 არის 0 ტიპის, ხოლო ატრაქციონი 3--1 ტიპის.



გვაქვს ორი საინტერესო სამეული:

- ტიპების სამეულისთვის $(0, 1, 0)$, შეგვიძლია ავირჩიოთ $(a_1, a_2, a_3) = (2, 3, 2)$.
- ტიპების სამეულისთვის $(1, 0, 1)$, შეგვიძლია ავირჩიოთ $(a_1, a_2, a_3) = (3, 2, 3)$.

ეს ნიშნავს რომ პირობა 1 დაკმაყოფილებულია

პროცედურას შეუძლია დააბრუნოს წყვილი $([0, 0, 0, 1], [(0, 1), (2, 3)])$. გაითვალისწინეთ, რომ ამ მაგალითისთვის მოცემული ამოხსნა შესაძლოა ოპტიმალური არ იყოს $K = 1$ -ისთვის.

სანიმუშო გრეიდერი

შეტანის ფორმატი:

```
K
```

გამოტანის ფორმატი:

```
N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

შევნიშნოთ, რომ სანიმუშო გრეიდერის გამოტანილი მონაცემები ემთხვევა მოთხოვნილ ფორმატს გამოტანის ფაილისთვის.