D. Laser Strike

Име на проблемот	Laser Strike
Временско ограничување	3 секунди
Мемориско ограничување	1 гигабајт

Атина и нејзината пријателка Софија неодамна откриле нова друштвена игра која им станала омилена: Laser Strike. Во оваа игра, двата играчи работат заедно за да отстранат N фигури од таблата. Играта се одвива во две фази. Штосот е во тоа што Софија нема да има целосни информации за играта. За да ја добијат играта, Атина и Софија мора да работат заедно, но воедно да комуницираат што е можно помалку.

На таблата има N уникатни фигури, нумерирани со целите броеви од 0 до $N-1$. И двата играчи можат да ги видат овие фигури. Исто така, постојат $N-1$ врски помеѓу парови од фигури, така што е возможно да се стигне до која било фигура од која било друга фигура следејќи ги овие врски. Со други зборови, овие врски формираат дрво. **Само Атина може да ги види овие врски; Софија не ги знае.**

Во првата фаза од играта, Атина одлучува за редослед $\ell_0, \ell_1, \ldots, \ell_{N-2}$ по кој треба да се отстрануваат фигурите, сè додека не остане само една. Овој редослед ќе се чува во тајност од Софија. Ако таа може да го повтори (реплицира), тие ќе ја добијат играта. Отстранувањето на фигури мора да го задоволува следново правило: секој пат кога ќе се отстрани фигура, таа мора да биде поврзана со точно една преостаната фигура. Со други зборови, отстранетата фигура мора да биде лист на дрвото формирано од преостанатите фигури и самата фигура. (Откако ќе се отстранат $N-1$-те фигури, последната фигура се отстранува автоматски и играчите победуваат.) Атина мора да избере редослед што одговара на горенаведеното правило.

Атина, исто така, ќе напише порака до Софија, во форма на бинарен стринг. Атина може да избере колку ќе е долга оваа порака – но колку е пократка пораката, толку повеќе поени ќе добијат.

После тоа, започнува втората фаза од играта. Целта на играта е Софија да отстрани $N-1$ фигури од таблата, по редоследот $\ell_0, \ell_1, \ldots, \ell_{N-2}$. Таа ќе преземе $N-1$ потези. Пред потегот i, Атина ѝ кажува на Софија пар од цели броеви a, b со следниве својства:

- $a < b$;
- сеуште постои пар од директно поврзани фигури со броевите a и b; и
- или a или b е точната фигура ℓ_i што треба да се отстрани во овој потег.

Да забележиме дека за Атина врската (a, b) е единствено (уникатно) одредена со листот ℓ_i во тековното дрво.

Потоа Софија ги отстранува или a или b од таблата. Ако ова била точната фигура – односно ℓ_i – тие продолжуваат да играат. Во спротивно, ја губат играта.

Ваша задача е да ги имплементирате стратегиите и на Атина и на Софија, така што тие ќе ја добијат играта.

Вашата програма ќе биде бодувана во зависност од должината на пораката што Атина ќе ја напише во првата фаза од играта.

Имплементација

Ова е проблем со повеќекратно извршување, со значење дека Вашата програма ќе се изврши два пати. Првиот пат кога ќе се изврши, треба да ја имплементира стратегијата на Атина за првата фаза од играта. После тоа, треба да ја имплементира стратегијата на Софија за втората фаза од играта.

Првата линија од влезот содржи два цели броја, P и N, каде што P е или 1 или 2 (прва или втора фаза), а N е бројот на фигури.

Следните влезни податоци зависат од фазата:

Фаза 1: Атина

По првата линија (опишана погоре), следните $N - 1$ линии од влезот го опишуваат дрвото. Секоја линија содржи два броја, a и b ($0 \leq a < b \leq N - 1$), што означуваат врска помеѓу фигурите a и b.

Вашата програма треба да започне со печатење на бинарен стринг со најмногу $1\,000$ знаци, пришто секој од нив е `0` или `1` - пораката напишана од Атина. Да забележиме дека за да генерира стринг со должина 0, програмата треба да отпечати празна линија.

После ова, програмата треба да отпечати $N - 1$ цели броеви $\ell_0, \ell_1, \ldots, \ell_{N-2}$ во посебни линии, што го означуваат редоследот по кој Атина сака да ги отстрани лисјата од дрвото. Редоследот мора да биде таков што ако фигурите се отстранат една по една од дрвото по овој редослед, отстранетата фигура секогаш мора да биде лист, т.е. дрвото секогаш мора да остане сврзано.

Фаза 2: Софија

По првата линија (опишана погоре), следната линија од влезот го содржи бинарниот стринг (пораката од Атина) од Фаза 1.

После ова, ќе има $N-1$ рунди на интеракција, по една за секој потег на Софија.

Во i-тиот потег, Вашата програма прво треба да прочита два броја, a и b ($0 \le a < b \le N-1$). Една од овие фигури е листот ℓ_i во редоследот на Атина, а другата фигура е единствената преостаната фигура поврзана со ℓ_i. Потоа, Вашата програма треба да го отпечати ℓ_i, што означува дека Софија го отстранува овој лист. Ако Вашата програма не го отпечати точниот лист ℓ_i, девојките ја губат играта и Вашето предадено решение ќе биде оценето со Wrong Answer за овој тест случај.

Детали

Доколку *збирот* од времињата на извршување на двете одделни извршувања на Вашата програма го надмине временското ограничување, Вашето предадено решение ќе се оцени со Time Limit Exceeded.

Осигурајте се дека го чистите (анг. flush) стандардниот излез по печатењето на секоја линија, во спротивно Вашата програма може да биде оценета со Time Limit Exceeded. Во Python, ова се случува автоматски сè додека користите `input()` за читање линии. Во C++, `cout << endl;` активира чистење покрај тоа што печати нов ред; ако пак користите printf, искористете `fflush(stdout);`.

Да забележиме дека правилното читање на празен стринг може да биде комплицирано. Зададените темплејти се справуваат со овој случај на правилен начин.

Ограничувања и бодување

- $N = 1\,000$.
- $0 \le a < b \le N-1$ за сите врски.

Вашето решение ќе биде тестирано на множество од тест групи, пришто секоја од нив носи одреден број поени. Секоја тест група содржи множество од тест случаи. За да ги добиете поените за дадена тест група, треба да ги решите сите тест случаи во таа тест група.

Група	Макс резултат	Ограничувања
1	8	Дрвото е ѕвезда. Тоа значи дека сите јазли освен еден се лисја.
2	9	Дрвото е линија. Тоа значи дека сите јазли освен два листа (листни јазли) имаат точно два соседни јазли.
3	21	Дрвото е ѕвезда од која излегуваат линии. Тоа значи дека сите јазли имаат еден или два соседни јазли, освен еден што има повеќе од два соседни јазли.
4	36	Растојанието помеѓу кои било два јазли е најмногу 10.
5	26	Без дополнителни ограничувања.

За секоја тест група што Вашата програма ќе ја реши точно, ќе добиете поени врз основа на следната формула:

$$\text{поени} = S_g \cdot (1 - 0.3 \cdot \log_{10} \max(K, 1)),$$

каде што S_g е максималниот резултат за тест групата, а K е максималната должина на порака на Атина потребна за кој било тест случај во таа тест група. **Вашиот резултат за секоја тест група ќе биде заокружен на најблискиот цел број.**

Табелата подолу го прикажува бројот на поени, за неколку вредности на K, што Вашата програма ќе ги добие ако ги реши сите тест групи со тоа K. Конкретно, за да постигнете резултат од 100 поени, Вашето решение мора да го реши секој тест случај со $K \leq 1$.

K	1	5	10	50	100	500	1000
Поени	100	79	70	49	39	20	11

Алатка за тестирање

За да го олесниме тестирањето на Вашето решение, ви обезбедивме едноставна алатка што можете да ја преземете. Видете ги „прилозите“ (анг. attachments) на дното од страницата со проблемите на системот Kattis. Алатката е опционална за употреба. Забележете дека официјалниот оценувач на Kattis се разликува од дадената алатка за тестирање.

За да ја искористите алатката, креирајте влезна датотека, како на пример „sample1.in“, која треба да започнува со еден број N проследен со $N - 1$ линии што го опишуваат дрвото, во ист формат како во Фаза 1. На пример, за примерот подолу:

```
7
0 1
1 2
2 3
0 4
0 6
1 5
```

За Python програми, да речеме `solution.py` (што вообичаено се извршува со `pypy3 solution.py`), извршете:

```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

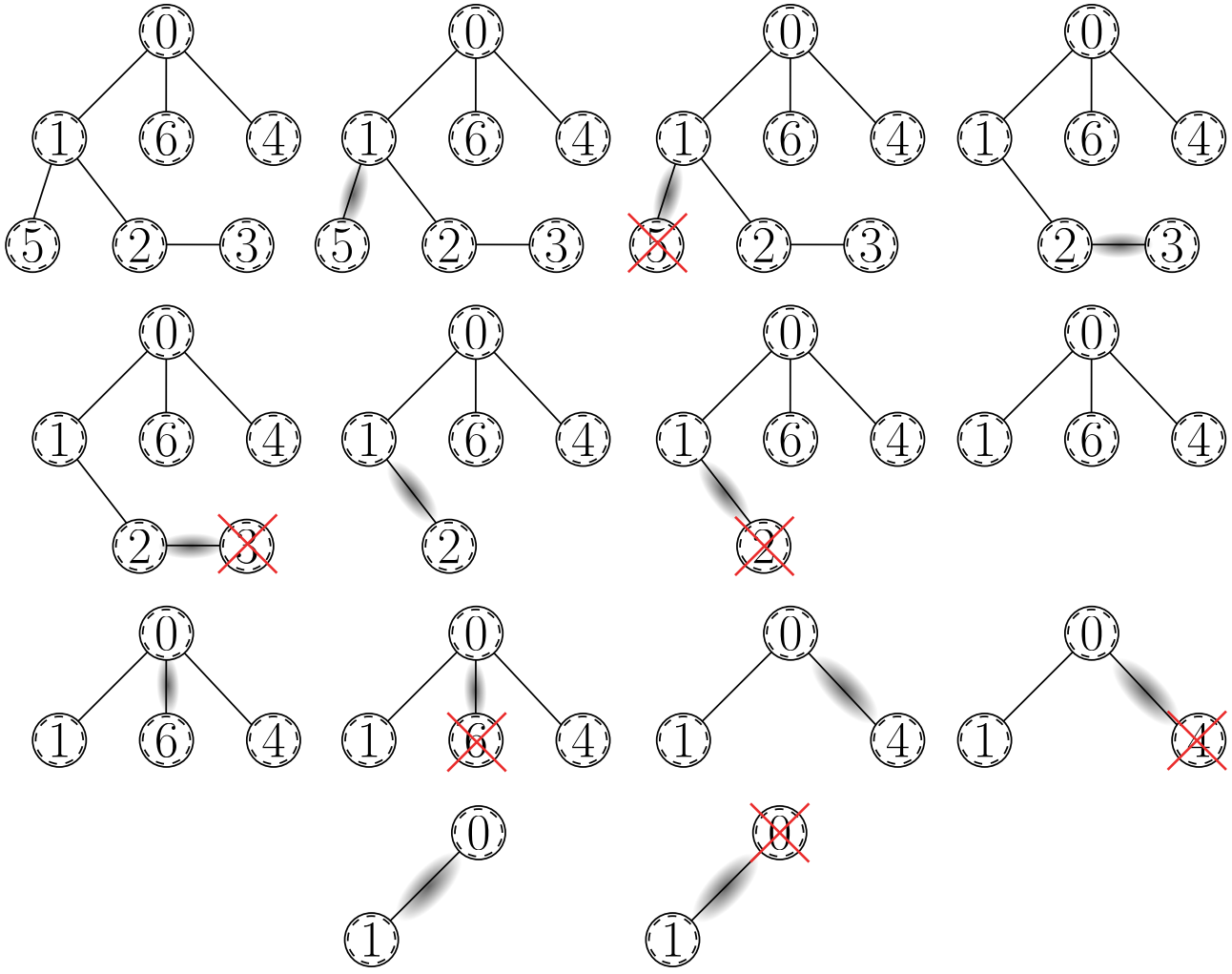
За C++ програми, прво искомпајлирајте (на пример, со `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`), а потоа извршете:

```
python3 testing_tool.py ./solution.out < sample1.in
```

Пример

Да забележиме дека во примерот од овој дел имаме $N = 7$ заради поедноставување, и затоа тој не е валиден тест случај. Не се очекува Вашата програма да може да го реши овој случај. Сите тест случаи на оценувачот ќе имаат $N = 1\,000$.

Во примерот, на Атина ѝ е дадено следното дрво. Во првата фаза, Атина го чита дрвото, избира бинарен стринг „`0110`“ за да го испрати до Софија, а исто така избира и редослед $[\ell_0, \ell_1, \ldots \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ по кој фигурите треба да се отстранат од дрвото. Во втората фаза, Софија го прима стрингот „`0110`“ што беше испратен во првата фаза. Потоа таа го добива парот $(1, 5)$ и одлучува да го отстрани темето 5, коешто всушност е листот. За следниот потег, таа го добива парот $(2, 3)$ и го отстранува листот 3, и така натаму. Следните слики ги прикажуваат интеракциите:


0 1 6 4 5 2 3
0 1 6 4 5 2 3
0 1 6 4 5 2 3
0 1 6 4 2 3
0 1 6 4 2 3
0 1 6 4 2
0 1 6 4 2
0 1 6 4
0 1 6 4
0 1 4
0 1
0 1

Излез од оценувачот	Ваш излез
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

Излез од оценувачот	Ваш излез
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0