

## D. Laser Strike

Problem Name	Laser Strike
Time Limit	3 seconds
Memory Limit	1 gigabyte

Ann și prietena ei Kathrin au descoperit recent un nou joc de societate care a devenit preferatul lor: Laser Strike. În acest joc, cei doi jucători lucrează împreună pentru a elimina  $N$  piese de pe tablă. Jocul se desfășoară în două faze. Problema este că Kathrin nu va avea informații complete despre joc. Pentru a câștiga jocul, Ann și Kathrin trebuie să lucreze împreună, comunicând cât mai puțin posibil.

Pe tablă există  $N$  piese unice, numerotate de la 0 la  $N - 1$ . Ambii jucători pot vedea aceste piese. De asemenea, există  $N - 1$  conexiuni între perechi de piese, astfel încât este posibil să se ajungă la orice piesă din oricare altă piesă urmând aceste conexiuni. Cu alte cuvinte, aceste conexiuni formează un arbore. **Doar Ann poate vedea aceste conexiuni; Kathrin nu le cunoaște.**

În prima fază a jocului, Ann decide asupra ordinii  $\ell_0, \ell_1, \dots, \ell_{N-2}$  în care piesele ar trebui eliminate, până când rămâne doar una.

Această ordine va fi ținută secret față de Kathrin. Dacă o poate reproduce, vor câștiga jocul. Eliminarea pieselor trebuie să respecte următoarea regulă: de fiecare dată când o piesă este eliminată, aceasta trebuie să fie conectată cu exact o piesă rămasă.

Cu alte cuvinte, piesa îndepărtată trebuie să fie o frunză a arborelui format din piesele rămase și din ea însuși. (După ce cele  $N - 1$  piese au fost eliminate, ultima piesă este eliminată automat, iar jucătorii câștigă.) Ann trebuie să aleagă o ordine care corespunde regulii de mai sus.

Ann va scrie un mesaj pentru Kathrin, sub forma unui șir binar. Ann poate alege cât de lung este acest mesaj – dar cu cât este mai scurt, cu atât primesc mai multe puncte.

Ulterior, începe a doua fază a jocului. Scopul jocului este ca Kathrin să elimine  $N - 1$  piese de pe tablă în ordinea  $\ell_0, \ell_1, \dots, \ell_{N-2}$ . Ea va face  $N - 1$  mutări. Înainte de mutarea  $i$ , Ann îi spune lui Kathrin o pereche de numere întregi  $a, b$  cu următoarele proprietăți:

- $a < b$ ;
- există încă o pereche de piese conectate direct cu numerele  $a$  și  $b$ ; și

- una dintre  $a$  și  $b$  este piesa corectă  $\ell_i$  care ar trebui eliminată la această mișcare. Rețineți că pentru Ann conexiunea  $(a, b)$  este determinată în mod unic de frunza  $\ell_i$  din arborele curent.

Apoi, Kathrin elimină fie  $a$  sau  $b$  de pe tablă. Dacă aceasta era piesa corectă – adică  $\ell_i$  – ei continuă să joace. Altfel, pierde jocul.

Sarcina ta este să implementezi strategiile ambelor părți, atât ale lui Ann, cât și ale lui Kathrin, astfel încât să câștige jocul.

Programul tău va fi punctat în funcție de lungimea mesajului pe care Ann îl scrie în prima fază a jocului.

## Implementation

Aceasta este o problemă multi-run, ceea ce înseamnă că programul va fi executat de două ori. La prima rulare, ar trebui să implementeze strategia lui Ann pentru prima fază a jocului. Apoi, ar trebui să implementeze strategia lui Kathrin pentru a doua fază a jocului.

Prima linie a datelor de intrare conține două numere întregi  $P$  și  $N$ , unde  $P$  este 1 sau 2 (prima sau a doua fază), iar  $N$  este numărul de piese.

Următoarele input-uri depind de fază:

### Phase 1: Ann

După prima linie (descrișă mai sus), următoarele  $N - 1$  linii ale datelor de intrare descriu arborele. Fiecare linie conține două numere  $a$  și  $b$  ( $0 \leq a < b \leq N - 1$ ), indicând o conexiune între piesele  $a$  și  $b$ .

Programul tău ar trebui să înceapă prin a afișa un șir binar cu cel mult 1 000 de caractere din 0 sau 1, mesajul scris de Ann.

Rețineți că, pentru a genera un șir de caractere cu lungimea 0, ar trebui să generezi o linie goală.

Apoi, ar trebui să afișeze  $N - 1$  numere întregi  $\ell_0, \ell_1, \dots, \ell_{N-2}$  pe linii separate, indicând ordinea în care Ann dorește să elimine frunzele arborelui. Ordinea trebuie să fie astfel încât, dacă piesele sunt eliminate una câte una din arbore în această ordine, piesa eliminată trebuie să fie întotdeauna o frunză, adică arborele trebuie să rămână întotdeauna conectat.

### Phase 2: Kathrin

După prima linie (descrișă mai sus), următoarea linie de intrare conține șirul binar (mesajul lui Ann) din Faza 1.

Apoi, vor exista  $N - 1$  runde de interacțiune, câte una pentru fiecare dintre mișcările lui Kathrin.

La a  $i$ -a mișcare, programul tău ar trebui să citească mai întâi două numere  $a$  și  $b$  ( $0 \leq a < b \leq N - 1$ ). Una dintre aceste piese este frunza  $\ell_i$  din ordinea lui Ann, iar cealaltă piesă este singura piesă rămasă conectată cu  $\ell_i$ . Apoi, programul tău ar trebui să afișeze  $\ell_i$ , indicând faptul că Kathrin elimină această frunză. Dacă programul tău nu afișează frunza corectă  $\ell_i$ , fetele pierd jocul, iar submisia ta va primi răspunsul Wrong Answer pentru acest test.

## Detalii

Dacă *suma* timpilor de execuție ale celor două runde separate ale programului depășește limita de timp, submisia trimisă va primi verdictul Time Limit Exceeded.

Asigurați-vă că faceți flush la standard output după afișarea fiecărei linii, altfel programul ar putea primi verdictul Time Limit Exceeded. În Python, acest lucru se întâmplă automat atâta timp cât folosești `input()` pentru a citi linii. În C++, `cout << endl;` face flush adițional cu afișarea unei linii noi; dacă folosești `printf`, folosește `fflush(stdout);`.

Rețineți că citirea corectă a unui șir gol poate fi dificilă. Șabloanele furnizate gestionează corect acest caz.

## Restricții și punctaj

- $N = 1\,000$ .
- $0 \leq a < b \leq N - 1$  pentru toate conexiunile.

Soluția dumneavoastră va fi testată pe un set de grupuri de teste, fiecare valorând un număr de puncte. Fiecare grup de teste conține un set de cazuri de testare. Pentru a obține punctele pentru un grup de teste, trebuie să rezolvați toate cazurile de testare din grupul de teste.

Grup	Scor maxim	Restricții
1	8	Arborele este o stea. Adică, toate nodurile, cu excepția unuia, sunt frunze.
2	9	Arborele este o linie. Adică, toate nodurile, cu excepția a două noduri frunză, au exact două noduri adiacente.
3	21	Arborele este o stea din care pornesc linii. Adică, toate nodurile au fie unul, fie două noduri adiacente, cu excepția unuia care are mai mult de două noduri adiacente.
4	36	Distanța dintre oricare două noduri este cel mult 10.
5	26	Fără restricții adiționale.

Pentru fiecare grup de teste pe care programul tău îl rezolvă corect, vei primi un scor bazat pe următoarea formulă:

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

unde  $S_g$  este scorul maxim pentru grupul de teste, iar  $K$  este lungimea maximă a mesajului lui Ann necesar pentru oricare dintre cazurile de testare din grupul de teste.

**Scorul pentru fiecare grup de teste va fi rotunjit la cel mai apropiat număr întreg.**

Tabelul de mai jos arată, pentru câteva valori ale lui  $K$ , numărul de puncte pe care programul dumneavoastră le va obține dacă rezolvă toate grupurile de teste cu acel  $K$ . În special, pentru a obține un scor de 100 puncte, soluția dumneavoastră trebuie să rezolve fiecare test cu  $K \leq 1$ .

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

## Testing Tool

Pentru a facilita testarea soluției dumneavoastră, v-am oferit un instrument simplu pe care îl puteți descărca. Consultați secțiunea „atașamente” din partea de jos a paginii Kattis. Folosirea instrumentul este opțional. Rețineți că programul oficial de evaluare de pe Kattis este diferit de instrumentul de testare.

Pentru a utiliza instrumentul, creați un fișier de intrare, de exemplui „sample1.in”, care ar trebui să înceapă cu numărul  $N$  urmat de  $N - 1$  linii care descriu arborele, în același format ca în Faza 1. De pildă, pentru exemplul de mai jos:

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

Pentru un program Python, spre exemplu `solution.py` (în mod normal se execută ca `pypy3 solution.py`), execută:

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

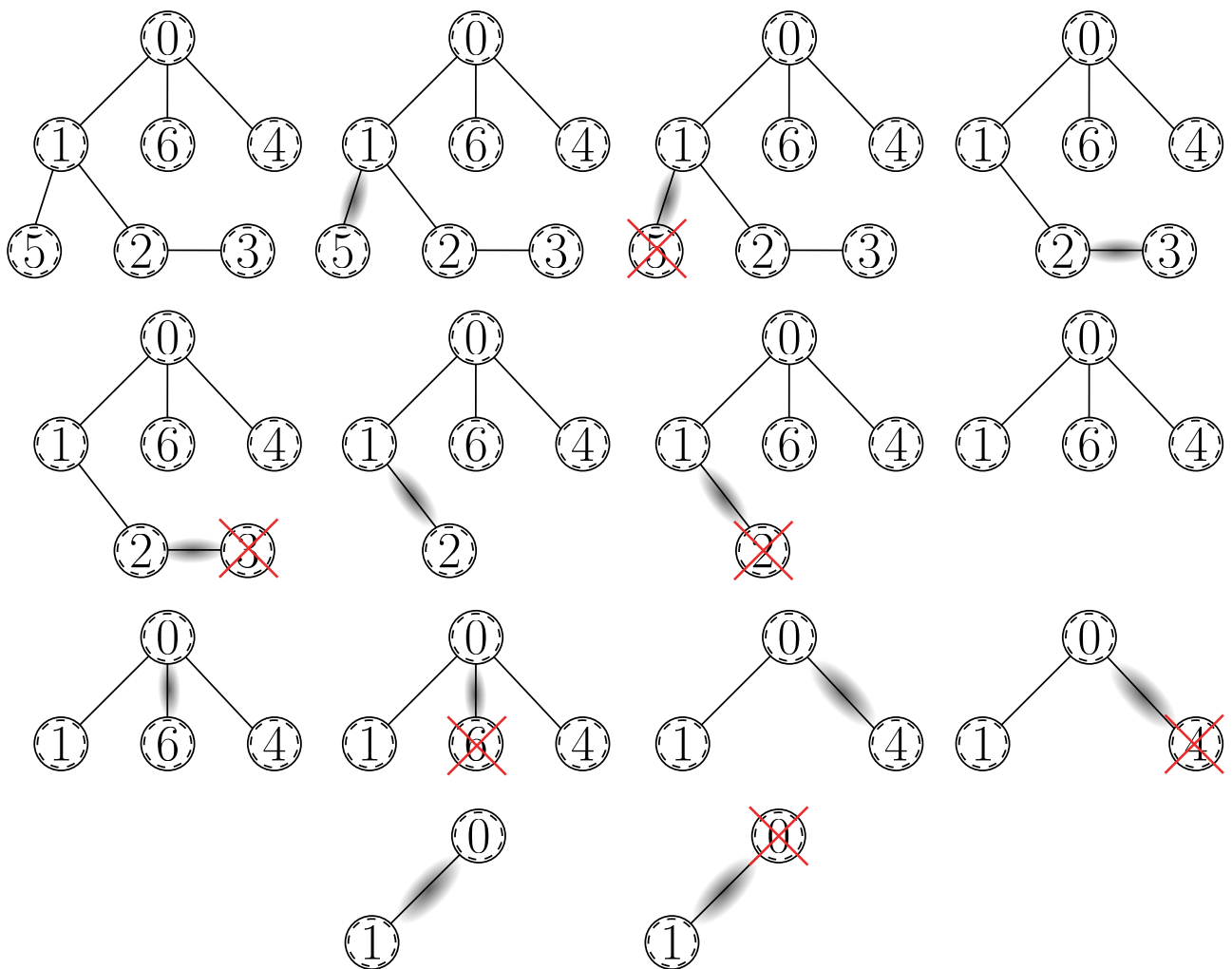
Pentru un program C++, inițial compilați-l (de exemplu cu `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) și apoi executați:

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

## Exemple

Rețineți că, exemplul din această secțiune are  $N = 7$  pentru simplitate și, prin urmare, nu este un caz de testare valid. Nu se așteaptă ca programul dumneavoastră să poată rezolva acest caz. Toate cazurile de testare din evaluator vor avea  $N = 1\,000$ .

În exemplu, lui Ann i se dă următorul arbore. În prima fază, Ann citește arborele, selectează un șir binar „0110” pentru a-l trimite către Kathrin și, de asemenea, selectează o ordine  $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$  în care piesele ar trebui eliminate din arbore. La a doua fază, Kathrin primește șirul de caractere „0110” care a fost trimis în prima fază. Apoi, ea primește perechea (1,5) și decide să elimine vârful 5, care este într-adevăr frunza. Pentru următoarea mutare, ea primește perechea (2,3) și îndepărtează frunza 3, apoi continuă la fel departe. Următoarele imagini ilustrează interacțiunile:



grader output	your output
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

grader output	your output
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0