

# Laser Strike (Drevorubačky)

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

Ema a Tamarka objavili novú doskovku. Volá sa Drevorubačky. Je to kooperatívna hra pre dve hráčky, ktoré sa spolu snažia dosiahnuť cieľ hry. Hlavnou pointou pravidiel je, že Tamarka nebude mať úplnú informáciu. Aby vyhrali, budú musieť spolupracovať, pričom cieľom hry je nielen vyhrať, ale navyše to dosiahnuť pomocou čo najmenšieho množstva komunikácie.

Na hracom pláne sa nachádza  $N$  navzájom rôznych guľičiek, očíslovaných od 0 po  $N - 1$ . Niektoré dvojice guľičiek sú medzi sebou prepojené paličkami. Paličiek je  $N - 1$  a vedú tak, aby všetko držalo pokope. Inými slovami, guľičky a paličky dokopy tvoria strom. **Strom vidí len Ema. Tamarka nevie, ako strom vyzerá.**

V prvej fáze hry si Ema musí zvoliť poradie  $\ell_0, \ell_1, \dots, \ell_{N-2}$  v ktorom bude zo stromu postupne odstránených niektorých  $N - 1$  guľičiek. (Tým už bude jednoznačne určená tá posledná, ktorá je na konci hry odstránená automaticky.) Toto poradie guľičiek bude pred Tamarkou utajené. Ak sa jej podarí dodržať ho, vyhrajú hru.

Ema musí zvoliť také poradie, ktoré zodpovedá nejakému možnému postupnému osekávaniu listov stromu. Musí teda platiť, že vždy, keď odstraňujeme nejakú guľičku, je táto guľička listom v strome, ktorý vznikol z pôvodného odstránením predchádzajúcich guľičiek (a paličiek vedúcich do nich). Inými slovami, odstraňovaná guľička vždy musí v strome susediť s práve jednou ešte neodstránenou guľičkou.

Po zvolení poradia môže ešte Ema zapísať správu, ktorú Tamarka neskôr dostane. Táto správa musí byť zapísaná ako postupnosť bitov. Ema si môže zvoliť, ako dlhú správu Tamarke pošle, ale na dĺžke záleží: čím kratšia správa, tým viac bodov je za výhru.

Následne začne druhá fáza hry. Na začiatku tejto fázy sa Tamarka dozvie Eminu správu. Následne bude Tamarka postupne hovoriť, v akom poradí treba odoberať guľičky zo stromu. Ak sa jej podarí zreprodukovat' Emou zvolené poradie  $\ell_0, \ell_1, \dots, \ell_{N-2}$ , hru vyhrajú.

Tamarka postupne spraví  $N - 1$  ťahov. Každý ťah začne tým, že Ema oznámi Tamarke dve celé čísla  $a$  a  $b$  s nasledovnými vlastnosťami:

- $a < b$ ,
- v strome ešte existuje dvojica guľičiek prepojených paličkou s číslami  $a$  a  $b$ ,
- jedna z guľičiek  $a$  a  $b$  je správna guľička  $\ell_i$ , ktorú má Tamarka odstrániť.

Všimni si, že Ema si v tomto kroku nič nevyberá: dvojica  $(a, b)$ , ktorú má oznámiť, je jednoznačne určená tým, že  $\ell_i$  je v liste aktuálneho stromu.

Keď sa Tamarka dozvie čísla  $a$  a  $b$ , musí si vybrať, ktorú z týchto dvoch guľičiek chce odstrániť zo stromu. Ak si vyberie správne, hra pokračuje. Ak sa pomýli, hra rovno končí a obe musia nahlas povedať: "Prehrala som."

Tvojou úlohou je implementovať obe stratégie tak, aby spolu vyhrali hru.

Body za korektný program dostaneš podľa toho, ako dlhú správu ti stačí poslať z prvej fázy do druhej.

## Implementácia

Pri testovaní bude tvoj program na každom teste postupne spustený dvakrát. Pri prvom spustení má hrať prvú fázu za Emu a pri druhom má hrať druhú fázu za Tamarku.

Pri oboch spusteniach sa v prvom riadku vstupu budú nachádzať dve celé čísla  $P$  a  $N$ .  $P$  je číslo fázy (1 alebo 2) a  $N$  je počet guľičiek.

Zvyšok vstupu je v každej fáze iný.

### Prvá fáza: Ema

V prvej fáze vstup pokračuje  $N - 1$  riadkami popisujúcimi jednotlivé hrany stromu. Každý z týchto riadkov obsahuje dve čísla  $a$  a  $b$  ( $0 \leq a < b \leq N - 1$ ) - čísla guľičiek prepojených paličkou.

Tvoj program by mal najskôr vypísať jeden riadok obsahujúci reťazec znakov 0 a 1: správu od Emy Tamarke. Tento reťazec môže mať od 0 po 1 000 znakov, vrátane. Ak chceš vypísať reťazec dĺžky 0, vypíš prázdny riadok.

Následne by tvoj program mal vypísať  $N - 1$  riadkov a v nich postupne po jednom čísle: čísla  $\ell_0, \ell_1, \dots, \ell_{N-2}$  určujúce poradie odstraňovania listov, ktoré si Ema zvolila. Vypísané poradie musí naozaj zopovedať nejakému platnému poradiu odstraňovania listov z daného stromu.

### Druhá fáza: Tamarka

Po rovnakom prvom riadku (obsahujúcom číslo fázy a počet guľičiek) je tento raz na vstupe riadok obsahujúci Eminu správu: reťazec núl a jednotiek, ktorý tvoj program vypísal počas prvej fázy.

Následne bude prebiehať  $N - 1$  kôl interakcie s testovačom: jedno pre každý Tamarkin ťah. Každé kolo musí prebehnúť nasledovne:

- Načítaš riadok s dvoma celými číslami  $a$  a  $b$  ( $0 \leq a < b \leq N - 1$ ). Jedno z týchto čísel je číslo správneho listu  $\ell_i$ , ktorý máš v tomto kole odstrániť. Druhé je číslo jedinej guľičky, ktorá je v tejto chvíli pripojená paličkou ku guľičke  $\ell_i$ .
- Určíš, či chceš odobrať guľičku  $a$  alebo guľičku  $b$ .
- Vypíšeš riadok s číslom guľičky, ktorú chceš odobrať.

Ak tvoj program nevypíše správne  $\ell_i$ , dievčatá prehrali a ty dostaneš za tento test verdikt Wrong Answer.

## Technické detaily

Ak *súčet* časov behu oboch spustení tvojho programu prekročí časový limit, dostaneš Time Limit Exceeded.

Po každom riadku vypísanom na výstup treba flushovať (čiže splachovať), inak ti tiež hrozí Time Limit Exceeded. Python flushuje automaticky, ak vstup načítavaš pomocou `input()`. V C++ príkaz `cout << endl;` vypíše znak nového riadku a následne flushne. Ak používaš `printf`, po každom z nich zavolaj `fflush(stdout);`.

Daj si pozor na to, že ak v prvej fáze vypíšeš prázdny reťazec, v druhej budeš musieť načítať riadok obsahujúci prázdny reťazec. Toto môže byť netriviálne. Ak si nie si istá, ako to spraviť, stiahni si a použi template riešenia, ktorý nachystali autori úlohy, ten to robí správne.

## Obmedzenia a hodnotenie

- V každom teste okrem príkladu platí  $N = 1\,000$ .
- To, či tvoje riešenie vyrieši vstup z príkladu správne, nemá vplyv na tvoje body.
- Pre každú dvojicu  $(a, b)$  platí  $0 \leq a < b \leq N - 1$ .

Za každú sadu vstupov dostaneš body, len ak ju celú vyriešiš správne. Počet bodov, ktorý za sadu dostaneš, je následne určený podľa kvality tvojich riešení.

Sada	Max body	Dodatočné obmedzenia
1	8	Strom je hviezda, t.j. všetky vrcholy okrem jedného sú listy.
2	9	Strom je cesta, t.j. dva vrcholy sú listy a všetky ostatné vrcholy majú práve dvoch susedov.
3	21	Strom je hviezda s cestami namiesto hrán. Teda platí, že všetky vrcholy okrem jedného majú stupeň nanajvýš 2.
4	36	Vzdialenosť ľubovoľných dvoch vrcholov stromu je nanajvýš 10.
5	26	

Za každú sadu, v ktorej všetko vyriešiš správne, dostaneš body podľa nasledovného vzorca:

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

V tomto vzorci  $S_g$  je maximálny počet bodov za danú sadu a  $K$  získame tak, že pre každý test v sade sa pozrieme na to, akú dlhú správu si si poslala z prvej fázy do druhej, a z týchto dĺžok zoberieme maximum.

Za každú sadu zvlášť tvoje skóre zaokrúhlime na najbližšie celé číslo.

Pre názornosť prikkladáme nasledujúcu tabuľku. V nej je pre niekoľko rôznych hodnôt  $K$  uvedené, koľko bodov dokopy dostane riešenie, ktoré každú sadu vyrieši správne, pričom najdlhšia správa, ktorú pri jej riešení použije, má dĺžku presne  $K$ .

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

Na získ plného počtu bodov (100) je teda potrebné každý test správne vyriešiť tak, že si pošleme správu dĺžky 0 alebo 1.

## Testing Tool

Aby sa ti ľahšie lokálne testovalo tvoje riešenie, môžeš si stiahnuť a použiť testing tool. Nájdeš ho v sekcii Attachments na spodku zadania tejto úlohy v Kattise. Používanie tohto nástroja je dobrovoľné. V skutočnom testovači sa na testovanie používa trochu iný skript, nie priamo tento.

Aby si mohla testing tool použiť, najskôr si na disku vyrob vstupný súbor, napr. nazvaný "sample1.in". Tento súbor by mal začať riadkom obsahujúcim  $N$  a následne by mal mať  $N - 1$  riadkov obsahujúcich popis hrán stromu vo formáte použitom v prvej fáze.

Príklad obsahu vstupného súboru:

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

Ak máš program písaný v Pythone v súbore `solution.py` (ktorý by si normálne spúšťala napr. príkazom `pypy3 solution.py`), môžeš testing tool použiť nasledovne:

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

Program v C++ si treba najskôr skompilovať (napr. príkazom `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) a potom môžeš testing tool použiť nasledovne:

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

## Príklad

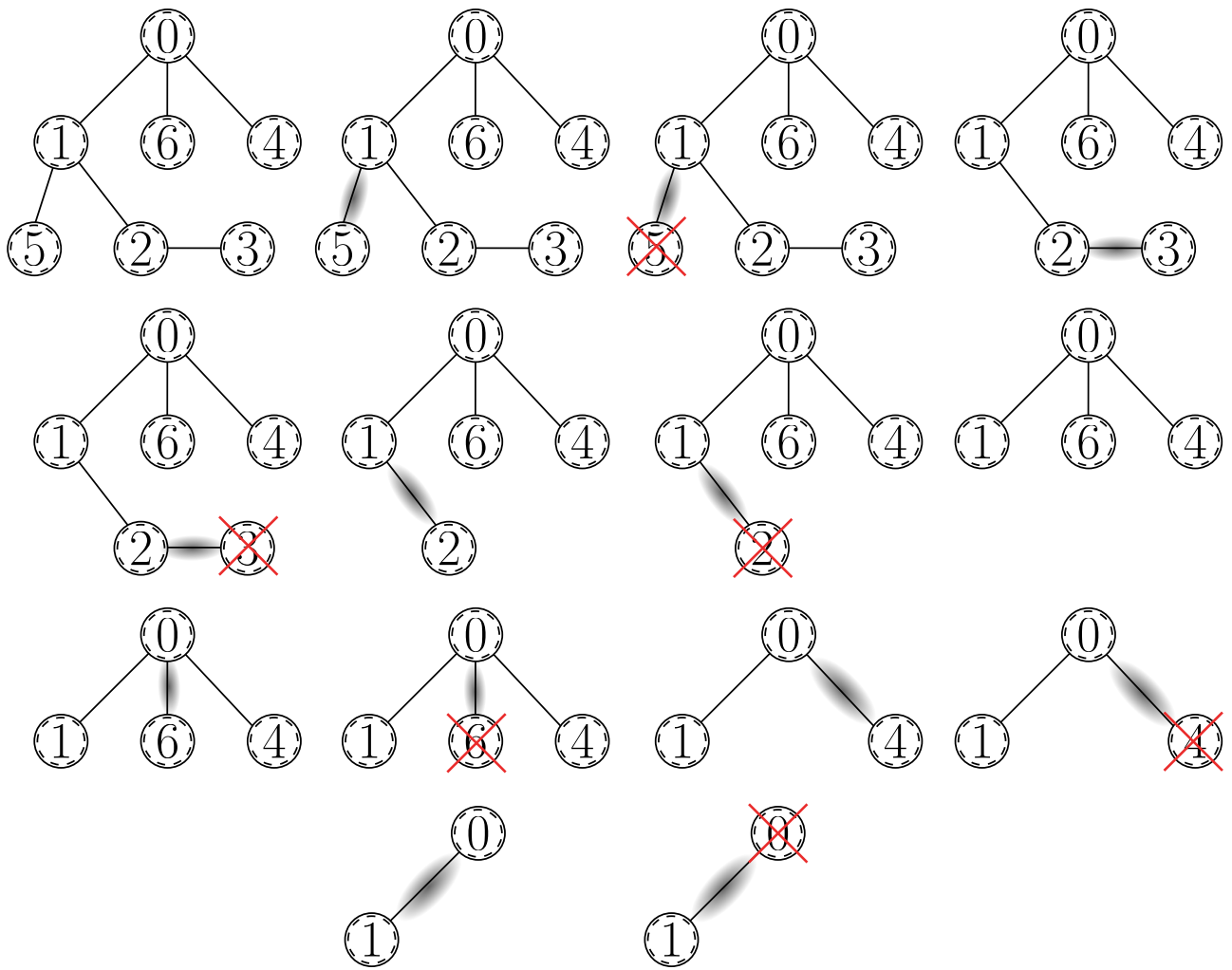
V tomto príklade Ema dostane strom zodpovedajúci vyššie uvedenému príkladu vstupného súboru.

V prvej fáze si Ema načíta popis stromu. Následne si zvolí a vypíše reťazec bitov pre Tamarku: "0110". Na záver vypíše konkrétne poradie, v ktorom má Tamarka odoberať listy zo stromu:  $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ .

V druhej fáze Tamarka dostane  $N$  a Emin reťazec "0110". Potom bude prebiehať interakcia:

- Tamarka načíta zo vstupu dvojicu (1, 5).
- Rozhodne sa odstrániť guľičku 5, toto vypíše na výstup.
- Keďže správne vypísala  $\ell_0$ , hra pokračuje.
- Tamarka načíta zo vstupu ďalšiu dvojicu (2, 3). Rozhodne sa odobrať guľičku 3 a vypíše jej číslo.
- a tak ďalej.

Celý priebeh interakcie je znázornený (postupne po riadkoch) na nasledujúcej sade obrázkov.



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