

D. Laser Strike

Problemnamn	Laser Strike
Tidsgräns	3 sekunder
Minnesgräns	1 gigabyte

Ann och hennes vän Kathrin har nyligen upptäckt det nya brädspelet Laser Strike som på kort tid har kommit att bli deras favorit. I Laser Strike arbetar de två spelarna tillsammans för att ta bort N pjäser från spelplanen. Spelet sker i två faser. Det som är spännande är att Kathrin inte kommer ha fullständig information om spelet. För att vinna spelet måste Ann och Kathrin samarbeta, men det ska ske med så lite kommunikation som möjligt mellan dem.

Det finns N unika pjäser på spelplanen, numrerade från 0 till $N - 1$. Båda spelarna kan se dessa pjäser. Det finns också $N - 1$ kopplingar mellan par av pjäser, som gör att det är möjligt att nå vilken pjäs som helst från vilken annan pjäs som helst genom att följa dessa kopplingar. Med andra ord bildar dessa kopplingar ett träd. **Endast Ann kan se dessa samband; Kathrin känner inte till dem.**

I spelets första fas bestämmer Ann en ordning $\ell_0, \ell_1, \dots, \ell_{N-2}$ enligt vilken pjäser ska tas bort, tills det bara finns en kvar. Denna ordning kommer hållas hemlig för Kathrin. Om hon sedan kan kopiera den kommer de att vinna spelet. Borttagningen av pjäser måste uppfylla följande regel: varje gång en pjäs tas bort måste den vara sammankopplad med exakt en återstående pjäs. Med andra ord måste den borttagna pjäsen vara ett löv från trädet som bildas av de återstående pjäserna och den själv. (Efter att $N - 1$ pjäser har tagits bort tas den sista pjäsen bort automatiskt och spelarna vinner.) Ann måste välja en ordning som motsvarar regeln ovan.

Ann kommer också att skriva ner ett meddelande till Kathrin, i form av en binär sträng. Ann kan välja hur långt meddelandet är – men ju kortare det är, desto fler poäng får de.

Efter det börjar spelets andra fas. Målet med spelet är att Kathrin ska ta bort $N - 1$ pjäser från spelplanen i ordningen $\ell_0, \ell_1, \dots, \ell_{N-2}$. Hon kommer göra $N - 1$ drag. Innan drag i säger Ann ett par av heltal a, b med följande egenskaper:

- $a < b$;
- det finns fortfarande två direkt sammankopplade pjäser med siffrorna a och b ; och
- antingen a eller b är den korrekta pjäsen ℓ_i som ska tas bort i detta drag.

Observera att för Ann bestäms kopplingen (a, b) unikt av lövet ℓ_i i det aktuella trädet.

Kathrin tar sedan bort antingen a eller b från spelplanen. Om detta var rätt pjäs – det vill säga ℓ_i – fortsätter de spela. Annars förlorar de spelet.

Din uppgift är att implementera både Anns och Kathrins strategier så att de vinner spelet.

Ditt program kommer att poängsättas beroende på längden av meddelandet som Ann skriver i spelets första fas.

Implementering

Ditt program kommer att köras två gånger. Första gången det körs ska ditt program implementera Anns strategi för spelets första fas. Andra gången ska det implementera Kathrins strategi för spelets andra fas.

Den första raden innehåller två heltal, P och N , där P är antingen 1 eller 2 (första eller andra fasen) och N är antalet pjäser.

Den följande indatan beror på vilken fas det är:

Fas 1: Ann

Efter den första raden (beskriven ovan) beskriver de följande $N - 1$ raderna trädet. Varje rad innehåller två siffror, a och b ($0 \leq a < b \leq N - 1$), som indikerar att det finns en koppling mellan pjäserna a och b .

Ditt program ska börja med att mata ut en binär sträng med högst 1 000 tecken, vardera 0 eller 1, som anger Anns meddelande. För att skriva ut en sträng med längden 0 kan du skriva en tom rad.

Efter detta ska programmet skriva ut de $N - 1$ heltalen $\ell_0, \ell_1, \dots, \ell_{N-2}$ på olika rader, för att ange vilken ordning Ann vill ta bort trädets löv i. Ordningen måste vara sådan att om pjäserna tas bort en efter en från trädet i denna ordning, kommer den borttagna pjäsen alltid vara ett löv, dvs. trädet måste alltid förbli sammanhängande.

Fas 2: Kathrin

Efter den första raden (beskriven ovan) innehåller nästa rad den binära strängen (Anns meddelande) från fas 1.

Efter detta kommer det finnas $N - 1$ interaktionsrundor, en för varje drag som Kathrin utför.

I det i :te draget ska ditt program först läsa in två tal, a och b ($0 \leq a < b \leq N - 1$). En av dessa pjäser är lövet ℓ_i i Anns ordning, och den andra pjäsen är den enda återstående pjäsen som är kopplad till ℓ_i . Därefter ska ditt program skriva ut ℓ_i , vilket innebär att Kathrin tar bort det lövet.

Om ditt program inte skriver ut rätt löv ℓ_i förlorar tjejerna spelet och din inskickning kommer bedömas som "Wrong Answer" för det testfallet.

Detaljer

Om *summan* av körtiderna för de två separata körningarna av ditt program överskrider tidsgränsen kommer din inskickning att bedömas som "Time Limit Exceeded".

Glöm inte att flusha standard output efter varje rad, annars kan ditt program bedömas som "Time Limit Exceeded". I Python sker det automatiskt så länge du använder `input()` för att läsa indatan. I C++ kommer `cout << endl;` både flusha och skriva ut en ny rad; om du använder `printf`, använd `fflush(stdout);`.

Begränsningar och poängsättning

- $N = 1\,000$.
- $0 \leq a < b \leq N - 1$ för alla kopplingar.

Din lösning kommer att testas på flera testgrupper, var och en värd ett antal poäng. Varje testgrupp innehåller en mängd testfall. För att få poäng för en testgrupp behöver du klara alla testfall inom gruppen.

Grupp	Maxpoäng	Begränsningar
1	8	Trädet är en stjärngraf. Det vill säga, alla noder utom en är lövnoder.
2	9	Trädet är en linje. Det vill säga, alla noder förutom två lövnoder har exakt två grannar.
3	21	Trädet är en stjärngraf med linjer som går ut från den. Dvs. alla noder har antingen en eller två grannar, förutom en som har mer än två grannar.
4	36	Avståndet mellan något par av noder är högst 10.
5	26	Inga ytterligare begränsningar.

För varje testgrupp som ditt program löser korrekt får du poäng enligt följande formel:

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

där S_g är den maximala poängen för testgruppen, och K är den maximala längden på Anns meddelande som behövs för något testfall i testgruppen. **Ditt resultat för varje testgrupp avrundas till närmsta heltal.**

Tabellen nedan visar antalet poäng som ditt program kommer att få för några värden på K , om det löser alla testgrupper med det K :et. I synnerhet, för att få 100 poäng måste din lösning lösa

varje testfall med $K \leq 1$.

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

Testverktyg

För att underlätta testningen av din lösning finns det ett enkelt verktyg som du kan ladda ner. Se "attachments" längst ner på problemsidan i Kattis. Det är valfritt att använda verktyget. Observera att den officiella gradern på Kattis skiljer sig från testverktyget.

För att använda verktyget, skapa en indatafil, till exempel "sample1.in", som ska börja med numret N följt av $N - 1$ rader som beskriver trädet, i samma format som i fas 1. Till exempel, för exemplet nedan:

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

För Python-program, exempelvis `solution.py` (körs normalt som `python3 solution.py`), ska du köra:

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

För C++-program, kompilera dem först (t.ex. med `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) och kör sedan:

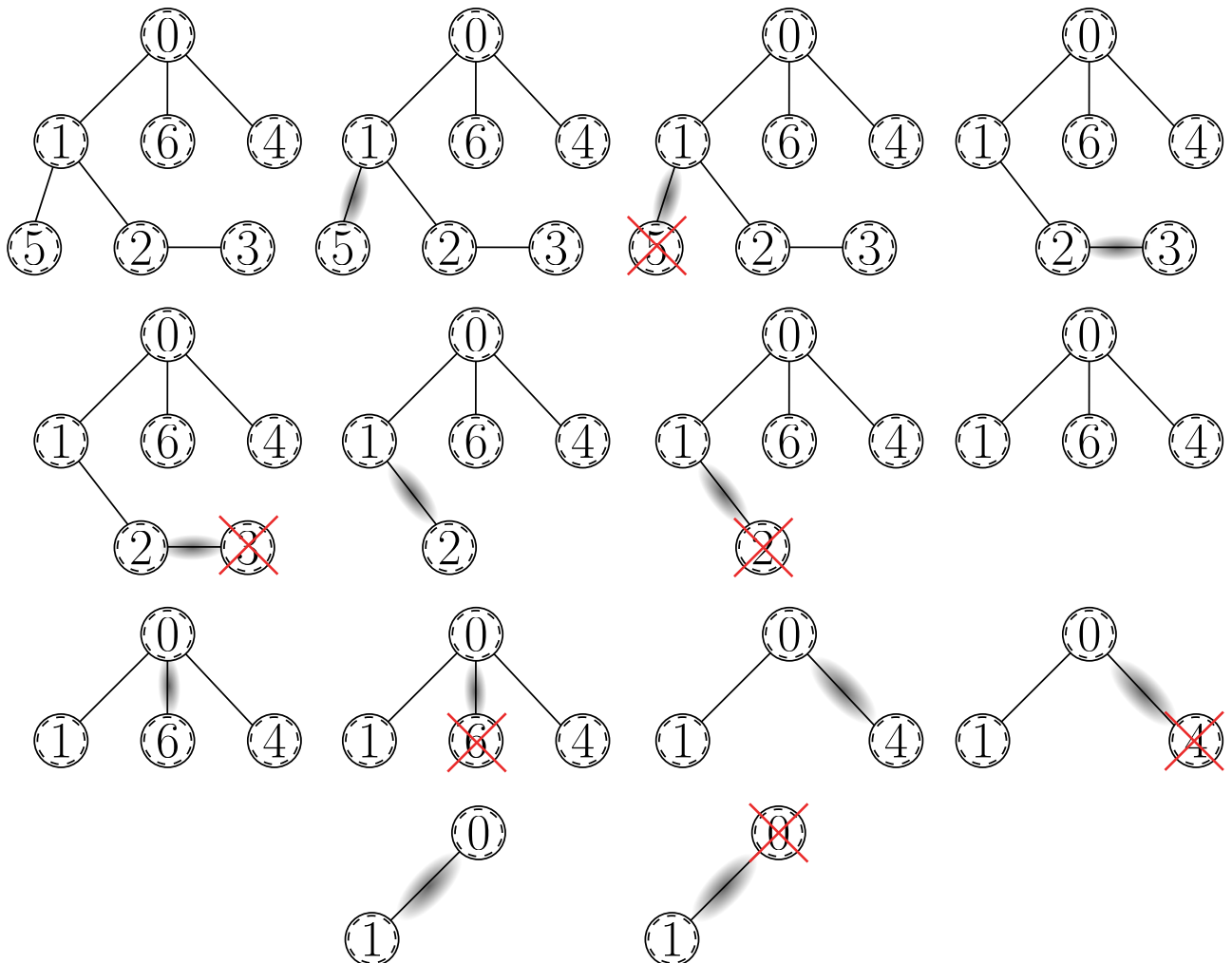
```
python3 testing_tool.py ./solution.out < sample1.in
```

Exempel

Observera att exemplet i detta avsnitt har $N = 7$ för enkelhetens skull och därför inte är ett giltigt testfall. Ditt program förväntas inte kunna lösa detta fall. Alla testfall på gradern kommer att ha $N = 1000$.

I exemplet får Ann följande träd. I den första fasen läser Ann in trädet, väljer en binär sträng "0110" att skicka till Kathrin, och väljer även en ordning $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ som

pjäserna ska tas bort från trädet. I den andra fasen tar Kathrin emot strängen "0110" som skickades i den första fasen. Hon får sedan paret (1, 5) och bestämmer sig för att ta bort nod 5, som faktiskt är lövet. I nästa drag får hon paret (2, 3) och tar bort lövet 3, och så vidare. Följande bilder visar interaktionerna:



graderns utdata	din utdata
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

graderns utdata	din utdata
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0