

D. Laser Strike

Nome del problema	Laser Strike
Limite di tempo	3 secondi
Limite di memoria	1 gigaoctetto

Ann e la sua amica Kathrin hanno scoperto di recente un nuovo gioco da tavolo che è diventato il loro preferito: Laser Strike. In questo gioco, i due giocatori lavorano insieme per rimuovere N pezzi dal tabellone di gioco. Il gioco si svolge in due fasi. Il problema è che Kathrin non avrà informazioni complete sulla partita. Per vincere la partita, Ann e Kathrin devono lavorare insieme, comunicando il meno possibile.

Ci sono N pezzi unici sul tabellone di gioco, numerati da 0 a $N - 1$. Entrambi i giocatori possono vedere questi pezzi. Esistono inoltre $N - 1$ connessioni tra coppie di pezzi, per cui è possibile raggiungere qualsiasi pezzo da qualsiasi altro pezzo seguendo queste connessioni. In altre parole, queste connessioni formano un albero. **Solo Ann può vedere queste connessioni; Kathrin non le conosce.**

Nella prima fase del gioco, Ann decide l'ordine $\ell_0, \ell_1, \dots, \ell_{N-2}$ in cui i pezzi devono essere rimossi, finché non ne rimane solo uno. Questo ordine rimarrà segreto a Kathrin. Se riuscirà a replicarlo, vinceranno la partita. La rimozione dei pezzi deve soddisfare la seguente regola: ogni volta che si rimuove un pezzo, questo deve essere collegato esattamente a un pezzo rimanente. In altre parole, il pezzo rimosso deve essere una foglia dell'albero formato dai pezzi rimanenti e da se stesso. (Dopo che sono stati rimossi i pezzi $N - 1$, l'ultimo pezzo viene rimosso automaticamente e i giocatori vincono.) Ann deve scegliere un ordine che corrisponda alla regola di cui sopra.

Ann scriverà anche un messaggio a Kathrin, sotto forma di stringa binaria. Ann può scegliere la lunghezza del messaggio: più è breve, più punti si ottengono.

Dopodiché inizia la seconda fase del gioco. L'obiettivo del gioco è che Kathrin rimuova $N - 1$ pezzi dalla scacchiera nell'ordine $\ell_0, \ell_1, \dots, \ell_{N-2}$. Farà dunque $N - 1$ mosse. Prima della mossa i , Ann comunica a Kathrin una coppia di numeri interi (a, b) con le seguenti proprietà:

- $a < b$;
- c'è ancora una coppia di pezzi direttamente collegati con numerati a e b ; e
- o a o b è il pezzo corretto ℓ_i che dovrebbe essere rimosso in questa mossa.

Si noti che per Ann la connessione (a, b) è determinata univocamente dalla foglia ℓ_i nell'albero corrente.

Kathrin rimuove quindi a o b dalla scacchiera. Se questo fosse il pezzo corretto, cioè ℓ_i , continuerebbero a giocare. Altrimenti perdono la partita.

Il tuo compito è mettere in pratica le strategie di Ann e Kathrin per far sì che vincano la partita.

Il punteggio del tuo programma dipenderà dalla lunghezza del messaggio che Ann scriverà nella prima fase del gioco.

Implementazione

Questo è un problema multi-esecuzione, il che significa che il programma verrà eseguito due volte. La prima volta che viene eseguito, dovrebbe implementare la strategia di Ann per la prima fase del gioco. La seconda volta dovrebbe implementare la strategia di Kathrin per la seconda fase del gioco.

La prima riga dell'input contiene due interi, P e N , dove P è 1 o 2 (prima o seconda fase), e N è il numero di pezzi.

Il seguente input dipende dalla fase:

Fase 1: Ann

Dopo la prima riga (descritta sopra), le successive $N - 1$ righe di input descrivono l'albero. Ogni riga contiene due numeri, a e b ($0 \leq a < b \leq N - 1$), che indicano una connessione tra i pezzi a e b .

Il programma dovrebbe iniziare stampando una stringa binaria con al massimo 1000 caratteri, ciascuno 0 o 1, il messaggio scritto da Ann. Si noti che per generare una stringa di lunghezza 0, dovrebbe essere visualizzata una riga vuota.

Dopodiché, dovrebbe stampare in output $N - 1$ interi $\ell_0, \ell_1, \dots, \ell_{N-2}$ su righe separate, indicando l'ordine in cui Ann desidera rimuovere le foglie dall'albero. L'ordine deve essere tale che se i pezzi vengono rimossi uno alla volta dall'albero in questo ordine, il pezzo rimosso deve essere sempre una foglia, cioè l'albero deve rimanere sempre connesso.

Fase 2: Kathrin

Dopo la prima riga (descritta sopra), la riga di input successiva contiene la stringa binaria (messaggio di Ann) della Fase 1.

Dopodiché ci saranno $N - 1$ round di interazione, uno per ogni mossa di Kathrin.

Nella i -esima mossa, il tuo programma dovrebbe prima leggere due numeri, a e b ($0 \leq a < b \leq N - 1$). Uno di questi pezzi è la foglia ℓ_i nell'ordine di Ann, e l'altro pezzo è l'unico pezzo rimanente connesso a ℓ_i . Quindi, il programma dovrebbe stampare ℓ_i , a indicare che Kathrin rimuove questa foglia. Se il programma non stampa la foglia corretta ℓ_i nell'ordine scelto da Ann, le ragazze perdono la partita e la tua soluzione verrà giudicata come Risposta Errata per questo caso di test.

Dettagli

Se la *somma* dei tempi di esecuzione delle due esecuzioni separate del programma supera il limite di tempo, la soluzione verrà giudicata come "Tempo Limite Superato".

Assicuratevi di flushare lo standard output dopo aver stampato ogni riga, altrimenti il vostro programma potrebbe essere giudicato come se avesse superato il limite di tempo. In Python, questo avviene automaticamente purché si usi `input()` per leggere le righe. In C++, `cout << endl;` esegue lo svuotamento oltre a stampare una nuova riga; se si usa `printf`, usare `fflush(stdout);`.

Si noti che leggere correttamente una stringa vuota può essere complicato. I modelli forniti gestiscono correttamente questo caso.

Assunzioni e punteggio

- $N = 1\,000$.
- $0 \leq a < b \leq N - 1$ per tutte le connessioni.

La tua soluzione verrà testata su una serie di subtask, ognuno dei quali varrà un certo numero di punti. Ogni subtask contiene una serie di casi di test. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

Subtask	Punteggio massimo	Assunzioni
1	8	L'albero è una stella. Cioè, tutti i nodi tranne uno sono foglie.
2	9	L'albero è una linea. Ovvero, tutti i nodi tranne due nodi foglia hanno esattamente due nodi adiacenti.
3	21	L'albero è una stella da cui partono delle linee. Ovvero, tutti i nodi hanno uno o due nodi adiacenti, tranne uno che ha più di due nodi adiacenti.
4	36	La distanza tra due nodi qualsiasi è al massimo 10.
5	26	Nessuna limitazione aggiuntiva.

Per ogni subtask che il tuo programma risolve correttamente, riceverai un punteggio basato sulla seguente formula:

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

dove S_g è il punteggio massimo per il subtask, e K è la lunghezza massima usata dal messaggio di Ann per ogni caso di test del subtask. **Il tuo punteggio per ciascun subtask verrà arrotondato al numero intero più vicino.**

La tabella seguente mostra il numero di punti, per alcuni valori di K , che il programma otterrà se risolve tutti i subtask con quel K . In particolare, per ottenere un punteggio di 100 punti, la tua soluzione deve risolvere ogni caso di test con $K \leq 1$.

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

Testing Tool

Per facilitare il test della tua soluzione, abbiamo messo a disposizione un semplice tool che puoi scaricare. Vedere gli "allegati" in fondo alla pagina del problema di Kattis. L'utilizzo del tool è facoltativo. Si noti che il grader di valutazione ufficiale su Kattis è diverso dallo tool di valutazione.

Per utilizzare il tool, creare un file di input, ad esempio "sample1.in", che dovrebbe iniziare con un numero N seguito da $N - 1$ righe che descrivono l'albero, nello stesso formato della Fase 1. Ad esempio, per il caso di esempio seguente:

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

Per i programmi Python, ad esempio `solution.py` (normalmente eseguito come `pypy3 solution.py`), eseguire:

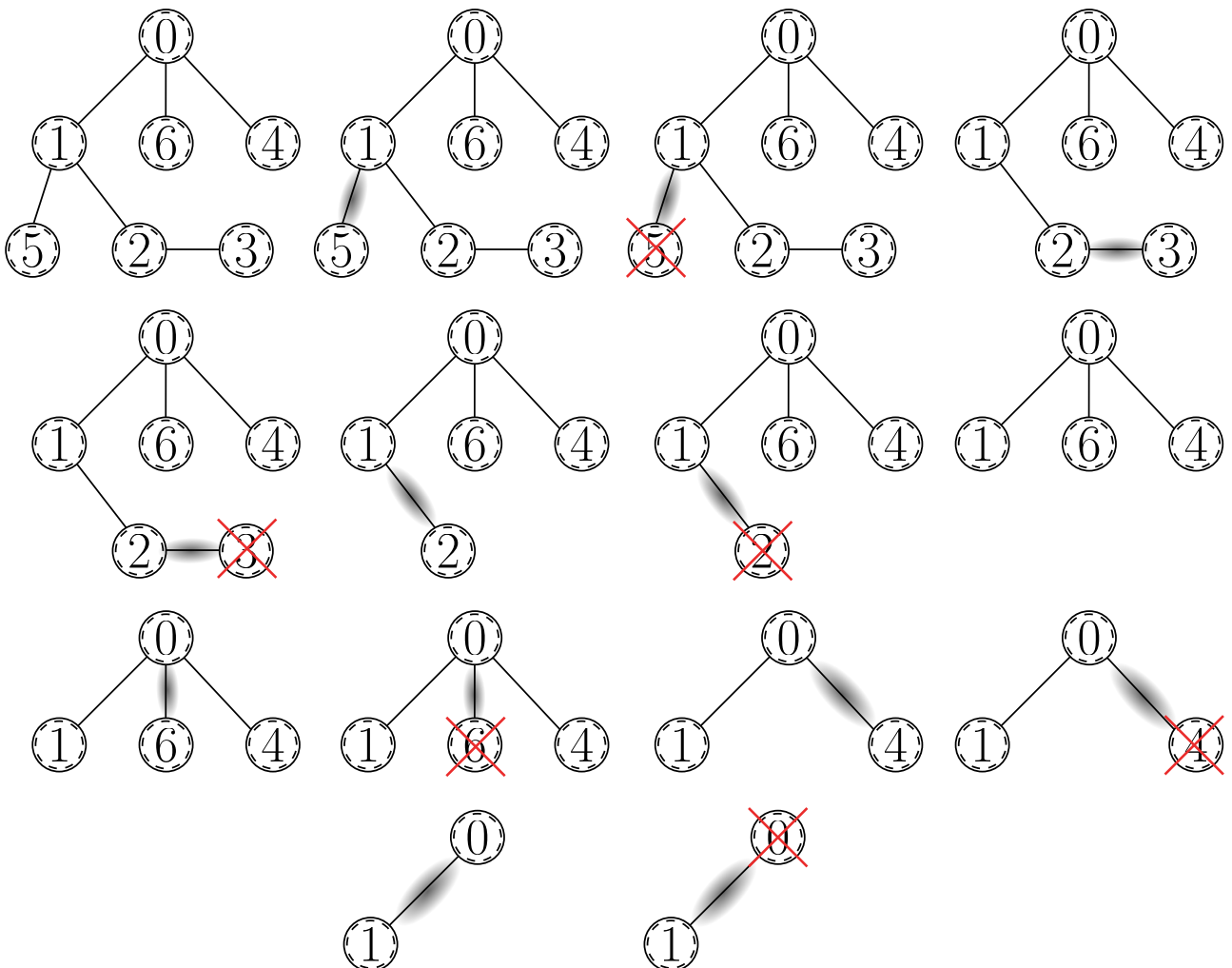
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Per i programmi C++, prima compilalo (ad esempio con `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) e poi esegui:

Esempio

Si noti che il caso di esempio in questa sezione ha $N = 7$ per semplicità e quindi non è un caso di test valido. Non ci si aspetta che il programma sia in grado di risolvere questo caso. Tutti i casi di test ufficiali avranno $N = 1\,000$.

Nell'esempio, ad Ann viene fornito il seguente albero. Nella prima fase, Ann legge l'albero, decide una stringa binaria " 0110 " da inviare a Kathrin e decide anche un ordine $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ in cui i pezzi devono essere rimossi dall'albero. Nella seconda fase, Kathrin riceve la stringa " 0110 " inviata nella prima fase. Riceve quindi la coppia $(1, 5)$ e decide di rimuovere il vertice 5, che è effettivamente la foglia. Per la mossa successiva, riceve la coppia $(2, 3)$ e rimuove la foglia 3, e così via. Le seguenti immagini illustrano le interazioni:



output del grader	il tuo output
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

output del grader	il tuo output
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0