

D. Ataque Laser

Nome do problema	Ataque de laser
Tempo limite	3 segundos
Memória limite	1 gigabyte

Ann e sua amiga Kathrin descobriram recentemente um novo jogo de tabuleiro que se tornou seu favorito: Laser Strike. Neste jogo, os dois jogadores trabalham juntos para remover N peças do tabuleiro. O jogo acontece em duas fases. O problema é que Kathrin não terá informações completas sobre o jogo. Para vencer o jogo, Ann e Kathrin precisam trabalhar juntas, comunicando-se o mínimo possível.

Há N peças únicas no tabuleiro, numeradas de 0 a $N - 1$. Ambos os jogadores podem ver essas peças. Também há $N - 1$ conexões entre pares de peças, de modo que é possível alcançar qualquer peça a partir de qualquer outra peça seguindo essas conexões. Em outras palavras, essas conexões formam uma árvore. **Somente Ann pode ver essas conexões; Kathrin não as conhece.**

Na primeira fase do jogo, Ann decide uma ordem $\ell_0, \ell_1, \dots, \ell_{N-2}$ na qual peças devem ser removidas até que reste apenas uma. Esta ordem será mantida em segredo de Kathrin. Se ela conseguir reproduzi-la, eles vencerão o jogo. A remoção de peças deve obedecer à seguinte regra: cada vez que uma peça for removida, ela deve ser conectada com exatamente uma peça restante. Em outras palavras, deve ser uma folha da árvore formada pelos pedaços restantes. (Após $N - 1$ peças serem removidas, a última peça é removida automaticamente e os jogadores ganham.) Ann deve escolher uma ordem que corresponda à regra acima.

Ann também escreverá uma mensagem para Kathrin, no formato de uma sequência binária. Ann pode escolher o tamanho da mensagem, mas quanto mais curta, mais pontos ela ganha.

Depois disso, a segunda fase do jogo começa. O objetivo do jogo é que Kathrin remova $N - 1$ peças do tabuleiro na ordem $\ell_0, \ell_1, \dots, \ell_{N-2}$. Ela fará $N - 1$ movimentos. Antes de mover i , Ann diz a Kathrin um par de inteiros a, b com as seguintes propriedades:

- $a < b$;
- ainda há um par de peças diretamente conectadas com os números a e b ; e
- a ou b é a peça correta ℓ_i que deve ser removida neste movimento.

Observe que, para Ann, a conexão (a, b) é determinada exclusivamente pela folha ℓ_i na árvore atual.

Kathrin então remove a ou b do tabuleiro. Se esta for a peça correta – isto é, ℓ_i – eles continuam tocando. Caso contrário, eles perdem o jogo.

Sua tarefa é implementar as estratégias de Ann e Kathrin para que elas ganhem o jogo.

Seu programa será pontuado dependendo do tamanho da mensagem que Ann escrever na primeira fase do jogo.

Implementação

Este é um problema de execução múltipla, o que significa que seu programa será executado duas vezes. Na primeira vez que for executado, ele deverá implementar a estratégia de Ann para a primeira fase do jogo. Depois disso, ele deve implementar a estratégia de Kathrin para a segunda fase do jogo.

A primeira linha da entrada contém dois inteiros, P e N , onde P é 1 ou 2 (primeira ou segunda fase), e N é o número de peças.

A entrada a seguir depende da fase:

Fase 1: Ann

Após a primeira linha (descrita acima), as próximas $N - 1$ linhas da entrada descrevem a árvore. Cada linha contém dois números, a e b ($0 \leq a < b \leq N - 1$), indicando uma conexão entre as partes a e b .

Seu programa deve começar gerando uma string binária com no máximo 1 000 caracteres, cada 0 ou 1, a mensagem escrita por Ann.

Depois disso, ele deve gerar $N - 1$ inteiros $\ell_0, \ell_2, \dots, \ell_{N-2}$ em linhas separadas, indicando a ordem em que Ann deseja remover as folhas da árvore. A ordem deve ser tal que, se as peças forem retiradas uma a uma da árvore nessa ordem, a peça retirada deve ser sempre uma folha, ou seja, a árvore deve permanecer sempre conectada.

Fase 2: Kathrin

Após a primeira linha (descrita acima), a próxima linha de entrada contém a sequência binária (mensagem de Ann) da Fase 1.

Depois disso, haverá $N - 1$ rodadas de interação, uma para cada movimento de Kathrin.

No i ésimo movimento, seu programa deve primeiro ler dois números, a e b ($0 \leq a < b \leq N - 1$). Uma dessas peças é a folha ℓ_i na ordem de Ann, e a outra peça é a única peça restante conectada a ℓ_i . Em seguida, seu programa deverá exibir ℓ_i , indicando que Kathrin removeu esta folha. Se o seu programa não exibir a folha correta ℓ_i , as meninas perdem o jogo e sua submissão será considerada como Resposta Errada.

Detalhes

Se a soma dos tempos de execução das duas execuções separadas do seu programa exceder o limite de tempo, seu envio será considerado como Tempo Limite Excedido.

Certifique-se de liberar a saída padrão após imprimir cada linha, caso contrário seu programa poderá ser considerado como Tempo Limite Excedido. Em Python, isso acontece automaticamente, desde que você use `input()` para ler linhas. Em C++, `cout << endl;` limpa, além de imprimir uma nova linha; se estiver usando `printf`, use `fflush(stdout);`.

Observe que ler corretamente uma string vazia pode ser complicado. Os modelos (templates) fornecidos lidam com esse caso corretamente.

Restrições e pontuação

- $N = 1\,000$.
- $0 \leq a < b \leq N - 1$ para todas as conexões.

Sua solução será testada em um conjunto de grupos de teste, cada um valendo um certo número de pontos. Cada grupo de teste contém um conjunto de casos de teste. Para obter os pontos de um grupo de teste, você precisa resolver todos os casos de teste no grupo de teste.

Grupo	Pontuação máxima	Restrições
1	8	A árvore é uma estrela. Ou seja, todos os nós, exceto um, são folhas.
2	9	A árvore é uma reta. Ou seja, todos os nós, exceto dois nós-folha, têm exatamente dois nós adjacentes.
3	21	A árvore é uma estrela com linhas partindo dela. Ou seja, todos os nós têm um ou dois nós adjacentes, exceto um que tem mais de dois nós adjacentes.
4	36	A distância entre quaisquer dois nós é no máximo 10.
5	26	Sem restrições adicionais.

Para cada grupo de teste que seu programa resolver corretamente, você receberá uma pontuação com base na seguinte fórmula:

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

onde S_g é a pontuação máxima para o grupo de teste, e K é o comprimento máximo da mensagem de Ann necessária para qualquer caso de teste no grupo de teste (limitado a pelo menos 1). Sua pontuação para cada grupo de teste será arredondada para o número inteiro mais próximo.

A tabela abaixo mostra o número de pontos, para alguns valores de K , que seu programa obterá se resolver todos os grupos de teste com esse K . Em particular, para atingir uma pontuação de 100 pontos, sua solução deve resolver todos os casos de teste com $K \leq 1$.

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

Ferramenta de teste

Para facilitar o teste da sua solução, fornecemos uma ferramenta simples que você pode baixar. Veja “anexos” na parte inferior da página do problema de Kattis. O uso da ferramenta é opcional. Observe que o programa de avaliação oficial do Kattis é diferente da ferramenta de teste.

Para usar a ferramenta, crie um arquivo de entrada, como “sample1.in”, que deve começar com um número N seguido por $N - 1$ linhas descrevendo a árvore, no mesmo formato da Fase 1. Por exemplo, para o exemplo abaixo:

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

Para programas Python, digamos `solution.py` (normalmente executado como `pypy3 solution.py`), execute:

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

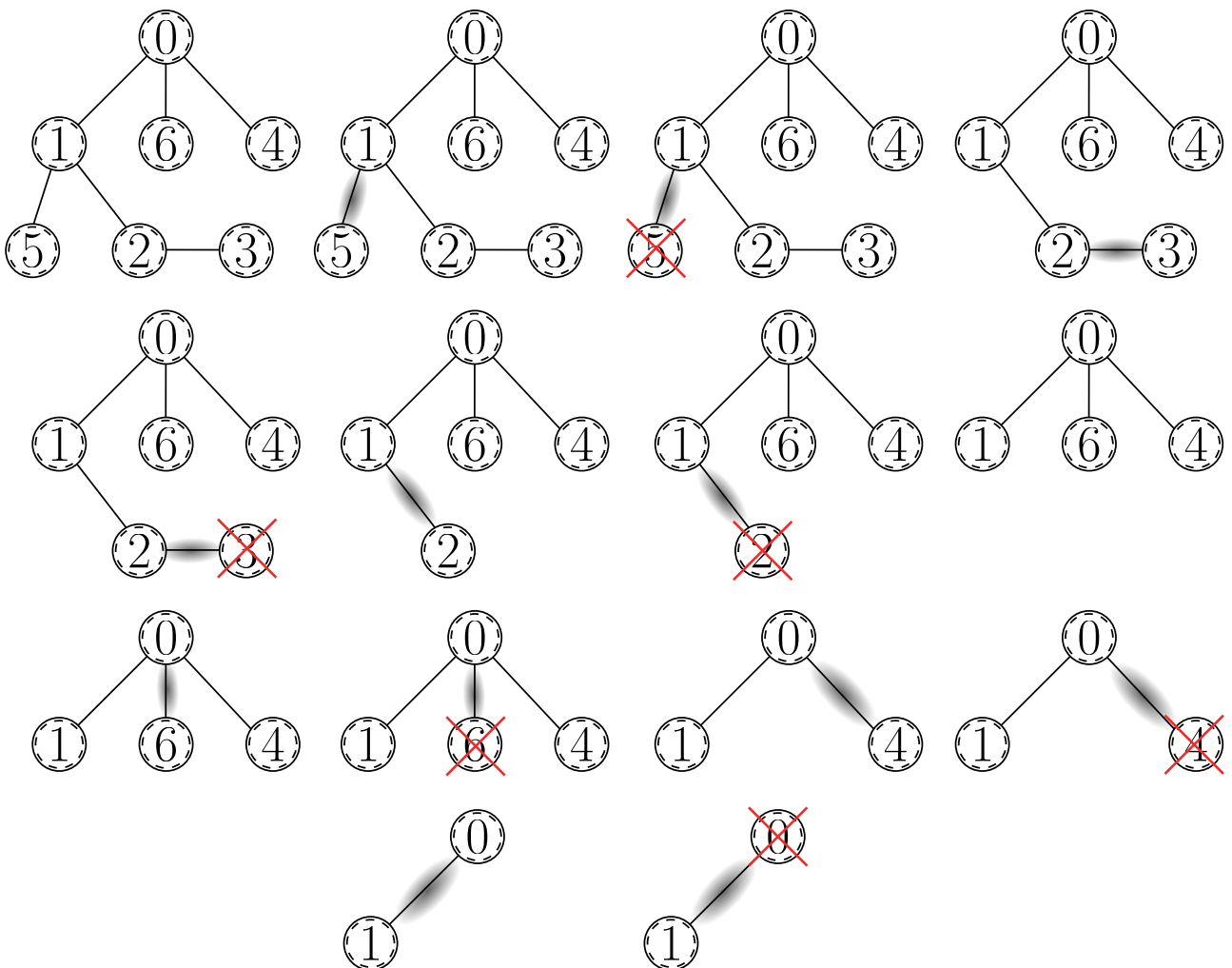
Para programas em C++, primeiro compile-o (por exemplo, com `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) e então execute:

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

Exemplo

Observe que o exemplo nesta seção tem $N = 7$ para simplificar e, portanto, não é um caso de teste válido. Não se espera que seu programa seja capaz de resolver este caso. Todos os casos de teste no classificador terão $N = 1\,000$.

No exemplo, Ann recebe a seguinte árvore. Na primeira fase, Ann lê a árvore, seleciona uma string binária " 0110 " para enviar a Kathrin e também seleciona uma ordem $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$ na qual as peças devem ser removidas da árvore. Na segunda fase, Kathrin recebe a string " 0110 " que foi enviada na primeira fase. Ela então recebe o par $(1, 5)$ e decide remover o vértice 5, que de fato é a folha. Para o próximo movimento, ela recebe o par $(2, 3)$ e remove a folha 3, e assim por diante. As imagens a seguir retratam as interações:



saída do classificador	sua saída
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

saída do classificador	sua saída
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0