



나는코더다 2019 송년대회 - Open Contest IamCoder Good-bye 2019 Open

December 14th, 2019

Sponsored by

Startlink



NEXON



NEXON

Problem A. 라면 사기 (Large)

입력 파일: standard input
출력 파일: standard output
시간 제한: 1 second
메모리 제한: 64 mebibytes

라면매니아 교준이네 집 주변에는 N 개의 라면 공장이 있다. 각 공장은 1번부터 N 번까지 차례대로 번호가 부여되어 있다.

교준이는 i 번 공장에서 정확하게 A_i 개의 라면을 구매하고자 한다($1 \leq i \leq N$).

교준이는 아래의 세 가지 방법으로 라면을 구매할 수 있다.

1. i 번 공장에서 라면을 하나 구매한다($1 \leq i \leq N$). 이 경우 비용은 B 원이 든다.
2. i 번 공장과 $(i+1)$ 번 공장에서 각각 라면을 하나씩 구매한다($1 \leq i \leq N-1$). 이 경우 비용은 $(B+C)$ 원이 든다.
3. i 번 공장과 $(i+1)$ 번 공장, $(i+2)$ 번 공장에서 각각 라면을 하나씩 구매한다($1 \leq i \leq N-2$). 이 경우 비용은 $(B+2C)$ 원이 든다.

최소의 비용으로 라면을 구매하고자 할 때, 교준이가 필요한 금액을 출력하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 라면 공장의 개수를 의미하는 자연수 N 과 두 자연수 B, C 가 사이에 공백을 두고 주어진다.
두번째 줄에 N 개의 정수 $A_1, \dots, A_N$ 가 사이에 공백을 두고 주어진다.

출력 형식

첫번째 줄에 교준이가 필요한 최소 금액을 출력한다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $3 \leq N \leq 10^6$
- $1 \leq B \leq 10^6$
- $1 \leq C \leq 10^6$
- $0 \leq A_i \leq 10^6$ ($1 \leq i \leq N$)

입력과 출력의 예

standard input	standard output
3 2 2 1 0 1	4
5 3 2 1 1 1 0 2	13

이 페이지는 의도적으로 비워 두었습니다.

Problem B. 분할하기

입력 파일: standard input
출력 파일: standard output
시간 제한: 0.5 seconds
메모리 제한: 512 mebibytes

집합 S 에 대하여, " $a, b \in S \implies (a|b) \in S$ "가 성립한다면 (또한 그러할 때에만) " S 가 좋은 집합이다"라고 정의하자. 여기서 '|'는 Bitwise-or를 의미한다.

예를 들어, $\{1, 3\}$ 은 좋은 집합이다. $(1|1) = 1$, $(1|3) = 3$, $(3|3) = 3$ 은 모두 $\{1, 3\}$ 의 원소기 때문이다. 다만, $(1|2) = 3$ 이므로 $\{1, 2\}$ 는 좋은 집합이 아니다.

전체 집합 $U = \{0, 1, \dots, 2^N - 1\}$ 의 부분집합 A 에 대하여, " A 의 원소의 개수가 K 고, A 과 $(U \setminus A)$ 가 모두 좋은 집합"이면 (또한 그러할 때에만) A 를 " N, K -분할"이라고 한다.

N 과 K 가 주어질 때, " N, K -분할"이 존재하는지 밝히고, 있다면 그러한 예를 출력하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 자연수 N 과 K 가 사이에 공백을 두고 주어진다.

출력 형식

만약 " N, K -분할"이 존재하지 않다면 첫번째 줄에 "NO"를 출력한다.

만약 " N, K -분할"이 존재한다면 첫번째 줄에 "YES"를 출력한다.

또한 두번째 줄에 " N, K -분할"인 집합의 원소를 오름차순으로 출력한다.

" N, K -분할"인 집합이 여러 개 존재한다면, 아무거나 출력해도 정답으로 인정된다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 12$
- $1 \leq K \leq 2^N$

입력과 출력의 예

standard input	standard output
1 1	YES 0
2 4	YES 0 1 2 3
3 3	YES 1 4 5
3 5	YES 0 2 3 6 7

이 페이지는 의도적으로 비워 두었습니다.

Problem C. 다오의 데이트

입력 파일: standard input
출력 파일: standard output
시간 제한: 1 second
메모리 제한: 512 mebibytes

12월인 오늘, 크레이지 파크의 버블힐에는 첫 눈이 내렸다.

새하얗고 예쁜 첫 눈을 디지니와 함께 보고 싶은 다오는, 눈 오는 것을 핑계로 디지니와 데이트를 하고자 한다.

버블힐은 세로 H 칸, 가로 W 칸인 격자 모양이다. 그중 몇몇 칸에는 블록이 놓여져 있기에 지나갈 수 없다.

위에서 i 번째 행, 왼쪽에서 j 번째 열의 칸의 위치를 (i, j) 라고 나타내자.

즉, 가장 왼쪽 위의 칸의 위치는 $(1, 1)$ 고, 가장 오른쪽 아래의 칸은 (H, W) 다.

현재, 다오와 디지니는 버블힐의 서로 다른 칸 위에 서 있다.

다오는 블록이 없고 상하좌우로 인접한 칸으로 이동할 수 있다. 다오는 이렇게 이동하면서 디지니에게 가고자 한다.

모든 것이 다오의 생각대로 흘러가면 좋겠지만, 마리드는 다오를 짝사랑하기에, 다오와 디지니의 만남을 가만히 보고만 있지 않을 것이다.

마리드는 다오와 디지니가 서로 만나 데이트를 하는 것을 원하지 않는다.

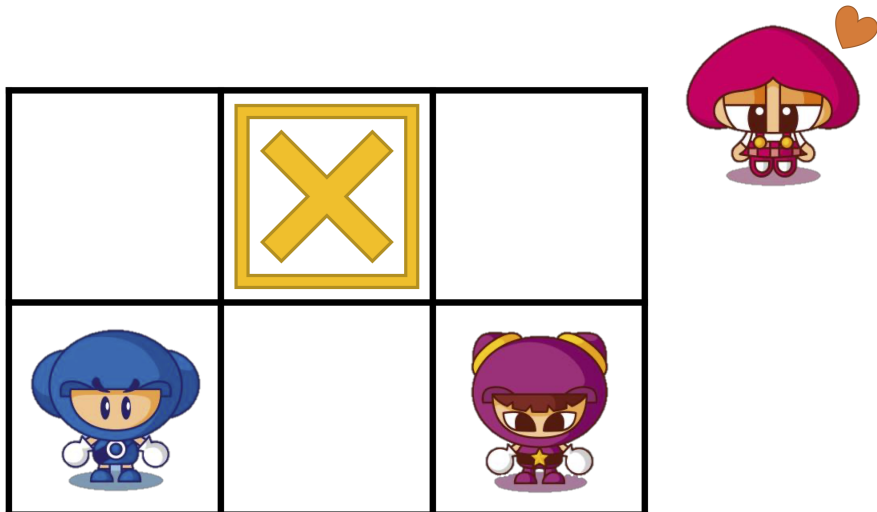
고로 마리드는 다오의 움직임을 방해하면서, 다오가 디지니를 만나지 못하도록 할 것이다.

마리드가 다오를 방해하는 방법은 다음과 같다.

1. 다오는 최대 N 번 움직일 수 있다.
2. 다오의 i 번째 움직임은 마리드가 정해진 두 방향 중 하나를 택하여, 그 방향을 향해야 한다.
예를 들어, 마리드가 “세번째 움직임을 왼쪽과 윗쪽 중 한 방향으로 해라”고 명령하면, 다오는 세번째로 움직일 때, 왼쪽으로 가거나 윗쪽으로 가야 한다.
만약 마리드가 제시한 두 방향 중 어느 방향으로도 다오가 갈 수 없다면, 다오는 더 이상 움직일 수 없다.

만약 버블힐이 다음과 같이 생겼다고 하자. $H = 2, W = 3, N = 2$ 다.

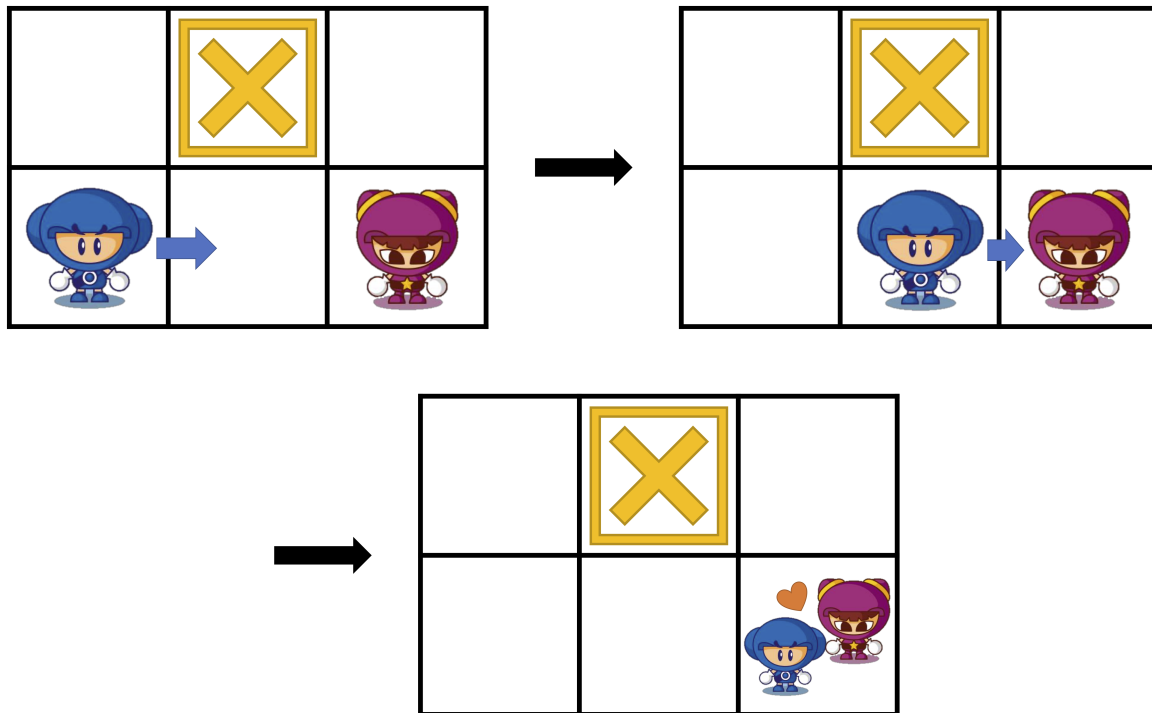
현재, 다오는 $(2, 1)$, 디지니는 $(2, 3)$ 에 위치해 있다. 또한 $(1, 2)$ 에는 블록이 있다.



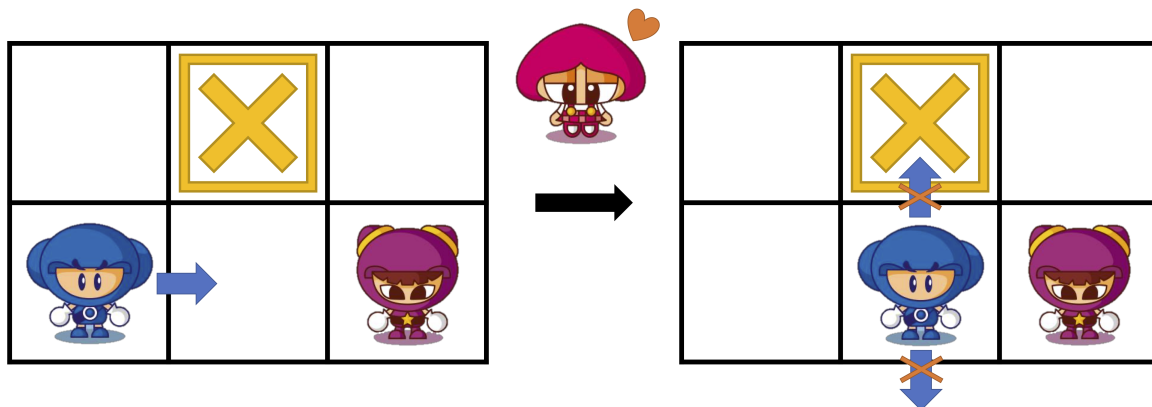
<그림 1> 다오와 디지니, 그리고 마리드

마리드의 방해가 없었다면, 다오는 오른쪽으로 두 번 이동해서 사랑하는 디지니를 만날 수 있다.
그러나 마리드가 “첫번째 움직임은 왼쪽과 오른쪽, 두번째 움직임은 왼쪽과 아래쪽 중 한 방향으로 해라”고 명령한다면, 다오는 디지니를 만날 수 없다. 다오가 첫번째 움직임으로 오른쪽을 택한다고 하더라도, 그 다음에 왼쪽이나 아래쪽으로 움직일 수 없기 때문이다.

블록이 있는 칸이나 버블힐 바깥으로는 이동할 수 없음에 유의하라.



<그림 2> 마리드가 없을 때, 행복한 다오와 디지니



<그림 3> 마리드로 인해 만날 수 없는 다오와 디지니

다오와 디지니는 서로를 너무 보고 싶어 한다.
버블힐의 정보와 마리드의 방해에 관한 정보가 전부 주어질 때, 다오가 디지니를 만날 수 있는지 판별하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 버블힐의 세로와 가로 칸 수를 의미하는 두 자연수 H 와 W 가 사이에 공백을 두고 주어진다.

두번째 줄부터 H 개의 줄에 걸쳐, 버블힐의 정보가 주어진다.

$(i+1)$ 번째 줄에는 길이 W 의 문자열이 주어진다($1 \leq i \leq H$). 이의 j 번째 문자를 $A_{i,j}$ 라고 하자($1 \leq j \leq W$). 이는 다음을 의미한다.

- $A_{i,j} = '.'$ 인 경우, 이는 (i, j) 에 아무것도 없음을 의미한다.
- $A_{i,j} = 'D'$ 인 경우, 이는 (i, j) 에 다오가 있음을 의미한다.
- $A_{i,j} = 'Z'$ 인 경우, 이는 (i, j) 에 디지니가 있음을 의미한다.
- $A_{i,j} = '@'$ 인 경우, 이는 (i, j) 에 블록이 있음을 의미한다.

$(H+2)$ 번째 줄에는 다오가 움직일 수 있는 최대 횟수를 의미하는 자연수 N 가 주어진다.

$(H+3)$ 번째 줄부터 N 개의 줄에 걸쳐, 마리의 방해에 관한 정보가 주어진다.

$(H+i+2)$ 번째 줄에는 두 문자 B_i 와 C_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq N$). 이는 다오가 i 번째 움직임으로 B_i 방향이나 C_i 방향 중 하나를 택해야 함을 의미한다.

각 문자가 의미하는 방향은 다음과 같다(키보드에서 “WASD”의 배치를 생각하라).

- ‘W’는 위쪽을 의미한다.
- ‘A’는 왼쪽을 의미한다.
- ‘S’는 아래쪽을 의미한다.
- ‘D’는 오른쪽을 의미한다.

출력 형식

만약 어떤 선택을 하더라도 다오가 디지니를 만날 수 없다면, 첫번째 줄에 “NO”를 출력한다.

만약 다오가 디지니를 만날 수 있다면, 첫번째 줄에 “YES”를 출력한다.

또한 두번째 줄에 다오가 어떻게 움직여야 하는지를 출력한다.

자세한 출력 형식은 입출력 예제를 참고하라.

다오가 디지니를 만날 수 있는 방법이 여러 가지라면, 그 중 아무거나 출력해도 정답으로 인정된다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $2 \leq H \leq 20$
- $2 \leq W \leq 20$
- $1 \leq N \leq 20$
- $A_{i,j}$ 는 '.', 'D', 'Z', '@' 중 하나다($1 \leq i \leq H, 1 \leq j \leq W$).
- $A_{i,j}$ 중 정확하게 하나는 'D'다($1 \leq i \leq H, 1 \leq j \leq W$).
- $A_{i,j}$ 중 정확하게 하나는 'Z'다($1 \leq i \leq H, 1 \leq j \leq W$).
- B_i 과 C_i 는 'W', 'A', 'S', 'D' 중 하나다($1 \leq i \leq N$).
- $B_i \neq C_i$ ($1 \leq i \leq N$)

입력과 출력의 예

standard input	standard output
2 3 .@. D.Z 2 W D W S	NO
2 3 .@. D.Z 2 A D A D	YES DD
2 3 .Z. .@D 3 W D W S A S	NO
2 3 .Z. .@D 3 W S A S W D	YES WA
2 2 @@ ZD 4 A D A D A D A D	YES ADA

부연 설명

첫번째 입출력 예제는 <그림 3>의 상황과 동일하다.

두번째 입출력 예제의 경우, 다오가 오른쪽으로 두 번 움직이면 디지니를 만날 수 있다.
또한 이것이 다오가 디지니를 만나는 유일한 방법이다.

세번째 입출력 예제의 경우, 다오는 어떻게 움직이더라도 디지니를 만날 수 없다.
만약 다오가 처음에 윗쪽으로 움직인다면, 두번째로 움직일 때에는 아랫쪽으로 갈 수밖에 없다.
왜냐하면 그때 다오는 (1,3)에 서 있어서, 더 이상 윗쪽으로 움직일 수 없기 때문이다.

네번째 입출력 예제의 경우, 다오는 처음에 윗쪽으로, 두번째에 왼쪽으로 이동해서 디지니를 만날 수 있다.
다오는 세 번까지 움직일 수 있었으나, 두 번만에 디지니를 만났기 때문에 더 이상 움직일 필요가 없었다.

마지막 입출력 예제의 경우, 다오는 한 번만에 디지니를 만날 수 있다.
그러나 디지니가 있는 칸을 지나쳤다가 다시 돌아와도 정답으로 인정된다.

이 페이지는 의도적으로 비워 두었습니다.

Problem D. 참 어려운 문제

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 512 mebibytes

N 개의 정점으로 이루어진 루트 있는 트리가 있다.

모든 정점에는 1번부터 N 번까지 번호가 붙어있다. 또한 i 번 정점은 A_i 번 색으로 칠해져 있다($1 \leq i \leq N$).

이 트리에는 아주 특이한 성질이 있다. 색깔이 같은 두 정점을 고르면, 이 둘은 항상 조상-자식 관계가 아니다.

교준이는 이 트리의 구조는 알지만, 루트가 어떤 정점인지는 알지 못한다. 루트가 될 수 있는 정점을 모두 찾는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 정점의 개수를 의미하는 자연수 N 이 주어진다.

두번째 줄에 정점의 색을 의미하는 N 개의 자연수 $A_1, \dots, A_N$ 가 사이에 공백을 두고 주어진다.

세번째 줄부터 $(N - 1)$ 개의 줄에 걸쳐, $(N - 1)$ 개의 간선의 정보가 주어진다.

$(i + 2)$ 번째 줄에는 i 번째 간선의 정보를 나타내는 두 자연수 B_i 와 C_i 가 사이에 공백을 두고 주어진다($1 \leq i < N$).

i 번째 간선은 B_i 번 정점과 C_i 번 정점을 서로 연결한다($1 \leq i < N$).

출력 형식

첫번째 줄에 루트가 될 수 있는 정점의 개수를 출력한다.

두번째 줄에 루트가 될 수 있는 정점의 번호의 합을 출력한다.

세번째 줄에 루트가 될 수 있는 정점의 번호의 제곱의 합을 출력한다.

어떠한 정점도 루트가 될 수 없다면, 이들의 값은 전부 0임에 유의하라.

제약 조건

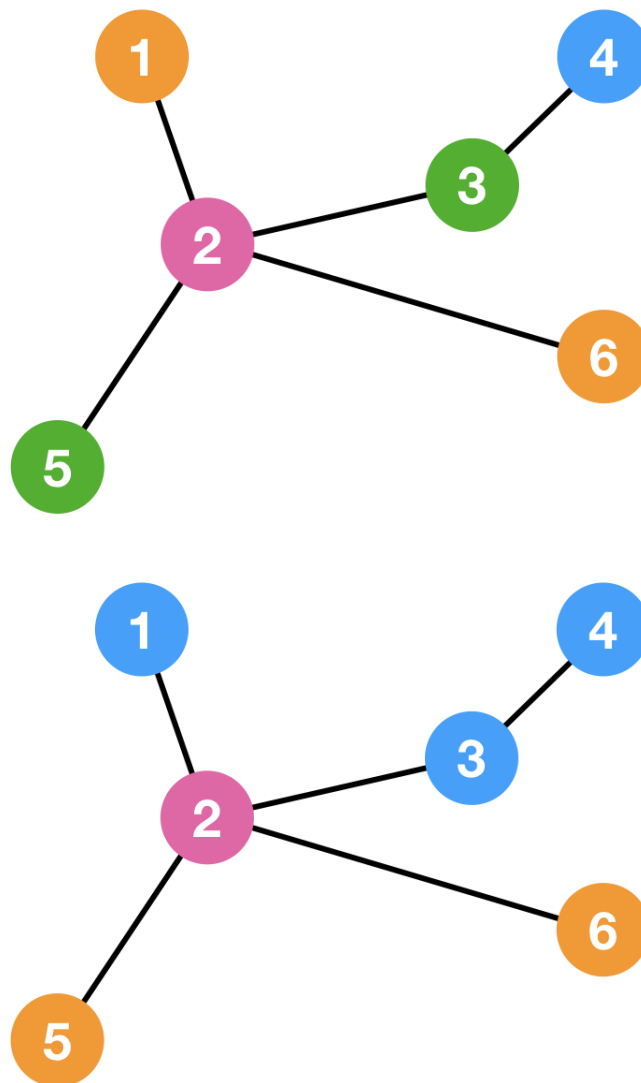
모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 250,000$
- $1 \leq A_i \leq N$ ($1 \leq i \leq N$)
- $1 \leq B_i \leq N$ ($1 \leq i < N$)
- $1 \leq C_i \leq N$ ($1 \leq i < N$)
- $B_i \neq C_i$ ($1 \leq i < N$)

입력과 출력의 예

standard input	standard output
6 3 2 4 1 4 3 1 2 2 3 2 5 2 6 3 4	1 2 4
6 1 2 1 1 3 3 1 2 2 3 2 5 2 6 3 4	0 0 0

부연 설명

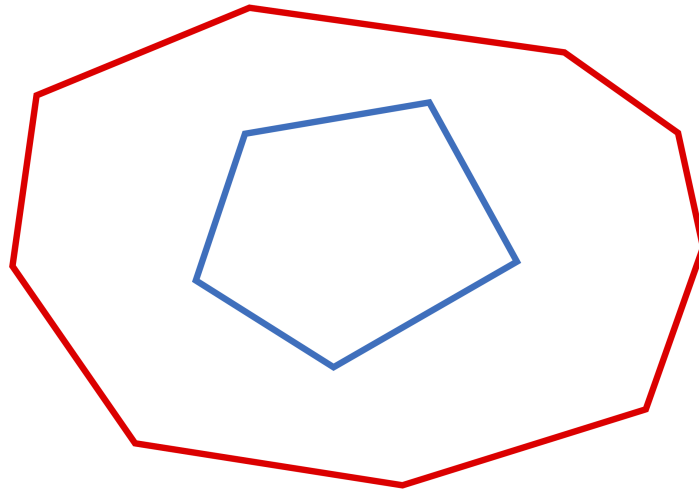


Problem E. 촛불과 그림자

입력 파일: standard input
출력 파일: standard output
시간 제한: 1 second
메모리 제한: 512 mebibytes

꼭짓점이 $(A_1, B_1), (A_2, B_2), \dots, (A_N, B_N)$ 인 빨간색 볼록 N 각형과, 꼭짓점이 $(C_1, D_1), (C_2, D_2), \dots, (C_M, D_M)$ 인 파란색 볼록 M 각형이 있다.

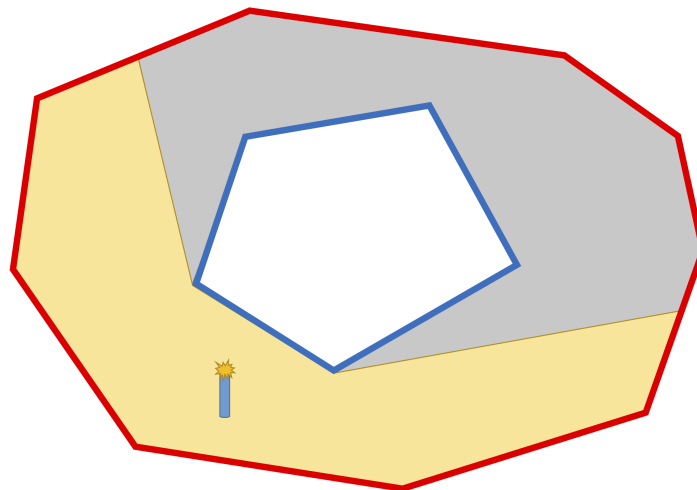
여기서 재미있는 점은 아래 그림과 같이, 빨간색 볼록 다각형 내부에 파란색 볼록 다각형이 완전하게 포함된다.



<그림 1> 파란색 다각형을 포함하는 빨간색 다각형

빨간색 볼록 다각형 내부이나, 파란색 볼록 다각형의 내부와 경계에 포함되지 않는 영역을 S 라 하자.

아래 그림과 같이, 영역 S 의 한 점에 촛불을 설치하면 노랑색 영역이 밝아지고, 회색 영역에는 그림자가 생긴다.



<그림 2> 밝혀진 영역과 그림자 영역

다각형의 정보와 촛불의 위치가 주어질 때, 그림자 영역의 넓이를 계산하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 빨간색 다각형과 파란색 다각형의 꼭짓점 개수를 의미하는 두 자연수 N 과 M , 쿼리의 개수를 의미하는 자연수 Q 가 사이에 공백을 두고 주어진다.

두번째 줄부터 N 개의 줄에 걸쳐, 빨간색 다각형의 꼭짓점 정보가 반시계 방향으로 주어진다.
($i + 1$)번째 줄에는 두 정수 A_i 와 B_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq N$).

($N + 2$)번째 줄부터 M 개의 줄에 걸쳐, 파란색 다각형의 꼭짓점 정보가 반시계 방향으로 주어진다.
($N + i + 1$)번째 줄에는 두 정수 C_i 와 D_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq M$).

($N + M + 2$)번째 줄부터 Q 개의 줄에 걸쳐, Q 개의 쿼리에 관한 정보가 주어진다.
($N + M + i + 1$)번째 줄에는 두 정수 X_i 와 Y_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq Q$). 이는 i 번째 쿼리에서는 촛불을 (X_i, Y_i) 에 설치함을 의미한다.

모든 Q 개의 쿼리는 서로 독립적임에 유의하라.

출력 형식

첫번째 줄부터 Q 개의 줄에 걸쳐, 쿼리에 대한 답을 차례대로 출력한다.

만약 촛불의 위치가 빨간색 볼록 다각형의 외부나 경계에 존재한다면, “OUT”을 출력한다.
만약 촛불의 위치가 파란색 볼록 다각형의 내부가 경계에 존재한다면, “IN”을 출력한다.
위 두 가지 경우가 모두 아니라면, 그림자의 영역의 넓이를 출력한다.

실수에 대한 절대·상대 오차는 10^{-6} 까지 허용된다.

제약 조건

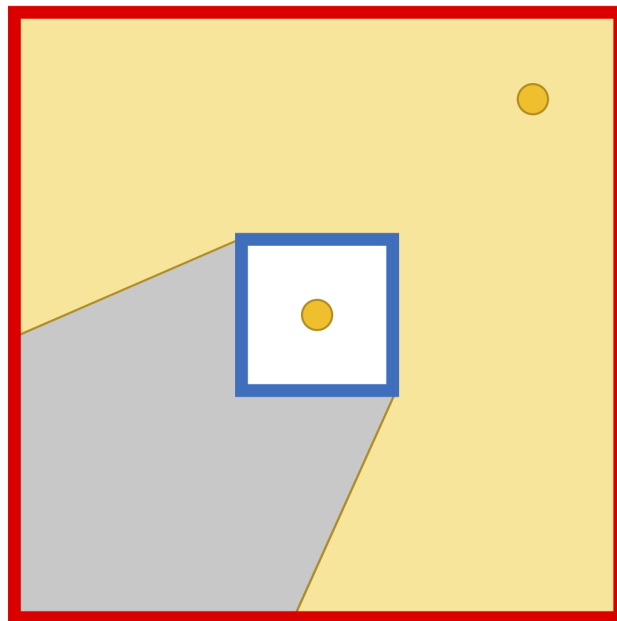
모든 입력 데이터는 다음 조건을 만족한다.

- $3 \leq N \leq 10^5$
- $3 \leq M \leq 10^5$
- $1 \leq Q \leq 10^5$
- $-10^6 \leq A_i \leq 10^6$ ($1 \leq i \leq N$)
- $-10^6 \leq B_i \leq 10^6$ ($1 \leq i \leq N$)
- $-10^6 \leq C_i \leq 10^6$ ($1 \leq i \leq M$)
- $-10^6 \leq D_i \leq 10^6$ ($1 \leq i \leq M$)
- $-10^6 \leq X_i \leq 10^6$ ($1 \leq i \leq Q$)
- $-10^6 \leq Y_i \leq 10^6$ ($1 \leq i \leq Q$)

입력과 출력의 예

standard input	standard output
4 4 3 -4 -4 4 -4 4 4 -4 4 -1 -1 1 -1 1 1 -1 1 0 0 3 3 6 6	IN 16.5 OUT

부연 설명



이 페이지는 의도적으로 비워 두었습니다.

Problem F. 피아노 연주

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 512 mebibytes

N 개의 손가락을 가진 세빈이는 한 손만으로 모든 곡을 연주할 수 있는 최고의 피아니스트다.
세빈이의 두 인접한 손가락 사이의 거리는 K 로 일정하다.
고로 i 번째 손가락과 첫번째 손가락 사이의 거리를 X_i 라고 한다면, $X_i = (i - 1)K$ 가 성립한다($1 \leq i \leq N$).

세빈이는 M 개의 음으로 이루어진 곡을 연주하려 한다. 이때 각각의 음을 어떤 손가락으로 연주하는지에 따라 곡을 연주하는 것이 쉬워질 수도, 어려워질 수도 있다.

i 번째 음의 음높이를 P_i 라 하고, 이 음을 F_i 번째 손가락으로 연주한다 하자. i 번째 음과 이와 인접한 $(i + 1)$ 번째 음을 연주할 때의 난이도는 $|(P_{i+1} - P_i) - (X_{F_{i+1}} - X_{F_i})|$ 다.

곡의 난이도는 인접한 두 음을 연주할 때의 난이도의 최댓값으로 정의한다. 각각의 음을 어떤 손가락으로 연주할지 결정하여 곡의 난이도를 최소화하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 세빈이의 손가락의 수와 곡을 이루는 음의 수를 나타내는 두 자연수 N 과 M , 두 인접한 손가락 사이의 거리를 나타내는 자연수 K 가 사이에 공백을 두고 주어진다.
두번째 줄에는 M 개의 정수 $P_1, \dots, P_M$ 이 사이에 공백을 두고 주어진다.

출력 형식

첫번째 줄에 곡의 난이도의 최솟값을 출력한다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 2 \times 10^5$
- $2 \leq M \leq 2 \times 10^5$
- $1 \leq K \leq 10^9$
- $X_N \leq 10^9$
- $1 \leq P_i \leq 10^9$ ($1 \leq i \leq M$)

입력과 출력의 예

standard input	standard output
5 4 3 7 2 9 3	1

부연 설명

$F_1 = 3, F_2 = 1, F_3 = 3, F_4 = 1$.

이 페이지는 의도적으로 비워 두었습니다.

Problem G. 보고 정렬

입력 파일: standard input
출력 파일: standard output
시간 제한: 4 seconds
메모리 제한: 1024 mebibytes

이 문제는 Interactive 문제입니다.

Bogo sort는 정렬이 될 때까지 랜덤 셔플하는 것을 반복하는 정렬 알고리즘이다.
이 문제에서 Grader는 0부터 $(N - 1)$ 까지의 정수로 이루어진 길이 N 의 순열 $A_0, \dots, A_{N-1}$ 를 가지고 있다.
여러분은 순열에서 어떤 연속한 구간을 잡아 랜덤 셔플하는 동작만을 반복하여 그 순열을 정렬해야 한다.
다행히, 랜덤 셔플이 어떻게 이루어졌는지 ‘보고’ 다음 동작을 결정할 수 있다.

함수 목록 및 정의

여러분의 프로그램은 제공되는 “bogoSort.h” 파일을 include 해야 한다.

여러분은 아래와 같은 함수를 구현해야 한다.

```
void sort_array(int N)
```

- N 은 그레이더가 가진 순열의 길이 N 을 의미한다.
- 이 함수는 프로그램이 실행될 때 정확히 한 번 호출된다.
- 이 함수가 종료될 때, 순열은 정렬되어 있어야 한다. 즉, 모든 $0 \leq i < N$ 에 대해 $A_i = i$ 가 성립해야 한다.

여러분은 아래와 같은 함수를 호출할 수 있다.

```
vector<int> copy_array()
```

- 이 함수는 Grader가 가진 순열 $A_0, \dots, A_{N-1}$ 를 반환한다.
- 여러분은 이 함수를 최대 2,500회 호출할 수 있다.

```
void shuffle_array(int s, int e)
```

- $0 \leq s \leq e < N$ 이 성립해야 한다.
- 이 함수가 호출되면 Grader는 순열의 구간 $[s, e]$ 를 랜덤 셔플한다.
- 여러분은 이 함수를 최대 2,500회 호출할 수 있다.

함수의 자세한 동작 과정은 Sample Grader를 참고하라.

제공되는 Sample Grader는 실제 채점에 활용되는 Grader와 다를 수 있음에 유의하라.
단, 채점에 사용되는 Grader의 `copy_array`, `shuffle_array` 함수는 Sample Grader와 동일하게 구현되어 있다.

입력 형식

Sample Grader는 다음과 같은 정보를 표준 입력을 통하여 읽어들이는다. 여러분은 어떠한 입력도 받으면 안된다.

첫번째 줄에 순열의 길이를 나타내는 자연수 N 이 주어진다.

두번째 줄에 N 개의 정수 $A_0, \dots, A_{N-1}$ 가 사이에 공백을 두고 주어진다.

출력 형식

Sample Grader는 다음과 같은 정보를 표준 출력을 통하여 출력한다. 여러분은 어떠한 출력도 하면 안된다.

여러분이 올바르게 순열을 정렬한 경우, Sample Grader는 첫번째 줄에 “Accepted”를 출력한다.

또한 두번째 줄에 두 함수 `copy_array`와 `shuffle_array`를 호출한 횟수를 각각 출력한다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 200$
- $0 \leq A_i < N$ ($0 \leq i < N$)
- $A_i \neq A_j$ ($0 \leq i < j < N$)

입력과 출력의 예

standard input	standard output
4 2 0 1 3	Accepted 2 4

부연 설명

다음은 위의 입출력 예제에 대해서, Sample Grader와 Interaction하는 과정을 나타낸 것이다.

여러분이 작성한 코드	Grader	설명
	$N = 4, A = \{2, 0, 1, 3\}$	Grader에서 N 과 $A_0, \dots, A_3$ 의 값을 읽어들이는다.
	<code>sort_array(4)</code> 호출	<code>sort_array</code> 함수는 초기에 단 한 번만 호출된다.
<code>copy_array()</code> 호출		
	<code>return {2, 0, 1, 3}</code>	순열 A 의 정보가 주어진다.
<code>shuffle_array(0, 3)</code> 호출		
	$A = \{1, 0, 3, 2\}$	순열 A 를 랜덤 셔플한다.
<code>copy_array()</code> 호출		
	<code>return {1, 0, 3, 2}</code>	현재 순열 A 의 정보가 주어진다.
<code>shuffle_array(0, 1)</code> 호출		
	$A = \{0, 1, 3, 2\}$	
<code>shuffle_array(2, 3)</code> 호출		
	$A = \{0, 1, 3, 2\}$	순열 A 가 섞이지 않을 수도 있다.
<code>shuffle_array(2, 3)</code> 호출		
	$A = \{0, 1, 2, 3\}$	
<code>sort_array(N = 4)</code> 종료		여러분은 순열 A 를 올바르게 정렬하였다.

위의 Interaction 과정은 실제 정해와 전혀 관련이 없으며, 단지 이해를 돕기 위해 만든 예시임에 유의하라.

Problem H. 비행기 타고 가요

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 1024 mebibytes

정보과학을 잘하는 사람만 거주할 수 있다는 국가, ‘경곽국’의 존재를 아는가?
경곽국은 총 N 개의 도시로 이루어져 있으며, 각각 1번부터 N 번까지 번호가 부여되어 있다.

경곽국에서 도시간의 이동수단은 비행기가 유일하다. 모든 도시쌍에 대하여 그 두 도시 사이를 경유하는 비행기가 존재한다. 다만, 비행기를 탑승하기 위해서는 ‘비행기 표’가 필요하다.

경곽국에서 구매할 수 있는 비행기 표는 총 M 개가 있으며, 각각 1번부터 M 번까지 번호가 부여되어 있다.
 i 번 표는 가격이 A_i 원이며, 비행기의 출발 도시의 번호가 B_i 이상 C_i 이하, 도착 도시의 번호가 D_i 이상 E_i 이하일 때에만 사용할 수 있다($1 \leq i \leq M$).

모든 M 개의 표는 일회용이며 한 번만 구매할 수 있다.

교준이의 집은 K 번 도시에 있다. 현재 교준이는 집에서 놀고 있다.

교준이가 집을 나와 X 번 도시로 가기 위해 구매해야 하는 표 값의 합의 최솟값을 W_X 라 하자($1 \leq X \leq N$).
단, 교준이는 집을 나오기 전에, 필요한 모든 비행기 표를 구매한다.

만일 어떻게 표를 사서 이동하더라도 X 번 도시에 이동할 수 없다면, $W_X = -1$ 로 정의한다.

$W_1, \dots, W_N$ 의 값을 모두 구하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 경곽국의 도시의 개수를 의미하는 자연수 N , 비행기 표의 총 개수를 의미하는 자연수 M , 교준이의 집이 있는 도시의 번호 K 가 사이에 공백을 두고 주어진다.

두번째 줄부터 M 개의 줄에 걸쳐, M 개의 비행기 표의 정보가 주어진다.

$(i + 1)$ 번째 줄에는 i 번 표의 정보를 나타내는 다섯 개의 정수 A_i, B_i, C_i, D_i, E_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq M$).

출력 형식

첫번째 줄에 N 개의 정수 $W_1, \dots, W_N$ 를 사이에 공백을 두고 차례대로 출력한다.

$W_K = 0$ 임에 유의하라.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 250,000$
- $1 \leq M \leq 250,000$
- $1 \leq K \leq N$
- $0 \leq A_i \leq 10^9$ ($1 \leq i \leq M$)
- $1 \leq B_i \leq C_i \leq N$ ($1 \leq i \leq M$)
- $1 \leq D_i \leq E_i \leq N$ ($1 \leq i \leq M$)

입력과 출력의 예

standard input	standard output
1 1 1 0 1 1 1 1	0
3 1 1 10 1 2 3 3	0 -1 10
5 2 1 5 2 3 4 5 5 1 5 2 2	0 5 -1 10 10

Problem I. Bad Hair Day와 기댓값

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 512 mebibytes

농부 존에게는 N 마리의 소가 있다. 소들의 키는 다양하며, 두 마리의 소의 키가 서로 같을 수 있다.

자신의 헤어스타일이 마음에 들지 않았던 소들은 일렬로 서서 서로의 헤어스타일을 확인해 주려고 한다. 각 소들은 자신의 오른쪽을 바라보면서 자신보다 키가 같거나 큰 소가 등장하기 전까지 나온 모든 소들의 헤어스타일을 확인할 수 있다.

더 정확하게는, 왼쪽에서 i 번째에 서 있는 소의 키를 H'_i 라 하면, i 번째 소는 다음 조건을 만족할 때 (또한 그러할 때에만) j 번째 소를 볼 수 있다.

- $i < j$ 고, $H'_i > H'_j$ 다.
- $i < k < j$ 고, $H'_i \leq H'_k$ 를 만족하는 k 가 존재하지 않는다.

소들이 N 마리 있으므로, 그들이 일렬로 설 수 있는 방법은 총 $N!$ 가지다.

농부 존은 헤어스타일을 확인할 수 있는 소들 쌍의 수의 기댓값이 궁금해졌다.

농부 존을 도와, 이 기댓값을 출력하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 소의 수를 나타내는 자연수 N 이 주어진다.

두번째 줄에 N 마리의 소의 키를 나타내는 N 개의 자연수 $H_1, \dots, H_N$ 가 사이에 공백을 두고 주어진다.

출력 형식

어떤 서로소이고 음이 아닌 두 정수 P, Q 가 존재하여, 헤어스타일을 확인할 수 있는 소들 쌍의 수의 기댓값이 $\frac{P}{Q}$ 와 같다고 하자.

이때, $P \equiv QX \pmod{10^9 + 7}$ 를 만족하는 0 이상 $(10^9 + 7)$ 미만의 정수 X 를 구하여 첫번째 줄에 출력한다.

조건을 만족하는 P, Q, X 는 반드시 존재한다. 또한 X 는 유일하게 존재함이 보장된다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq H_i \leq 10^9$ ($1 \leq i \leq N$)

입력과 출력의 예

standard input	standard output
4 1 1 2 2	333333337
4 10 20 30 40	416666672

부연 설명

첫번째 입출력 예제에서, 소들이 일렬로 설 수 있는 방법과 헤어스타일을 확인할 수 있는 소들 쌍의 수는 아래 표와 같다.

소들이 일렬로 선 방법	헤어스타일을 확인할 수 있는 소 쌍의 수
1, 1, 2, 2	0
1, 2, 1, 2	1
1, 2, 2, 1	1
2, 1, 1, 2	2
2, 1, 2, 1	2
2, 2, 1, 1	2

따라서 기댓값은 $\frac{0 + 1 + 1 + 2 + 2 + 2}{6} = \frac{4}{3}$ 다.

Problem J. 정점 찾기

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 512 mebibytes

이 문제는 Interactive 문제입니다.

정점 N 개로 이루어진 그래프에 단 하나의 간선이 숨겨져 있다.
이 간선은 A 번 정점과 B 번 정점을 연결하는 양방향 간선이다.
숨겨진 간선이 연결하는 두 정점의 번호 A, B 를 찾아내는 프로그램을 작성하시오.

함수 목록 및 정의

여러분의 프로그램은 제공되는 “findVertices.h” 파일을 include 해야 한다.

여러분의 프로그램은 한 번 실행될 때마다 T 개의 테스트 케이스를 처리해야 한다. 프로그램의 실행 시간은 주어진 모든 테스트 케이스를 처리하는 데 걸린 시간으로 측정한다.

여러분은 아래와 같은 함수를 구현해야 한다.

```
void find(int N)
```

- N 은 그래프의 정점의 개수를 의미한다.
- 이 함수는 테스트 케이스마다 정확히 한 번 호출된다.

여러분은 아래와 같은 함수를 호출할 수 있다.

```
int query(vector<int> U, vector<int> V, int p, int q)
```

- 두 배열 U, V 에 대응되는 수열을 U, V 라 하자.
- 수열 U, V 의 길이는 M 으로 동일해야 하며, $0 \leq M \leq 5,000$ 을 만족해야 한다.
- 모든 $0 \leq i < M$ 에 대해, $1 \leq U_i \leq N, 1 \leq V_i \leq N, U_i \neq V_i$ 을 만족해야 한다.
- 모든 $0 \leq i < j < M$ 에 대해, $(U_i, V_i) \neq (U_j, V_j)$ 와 $(U_i, V_i) \neq (V_j, U_j)$ 를 만족해야 한다.
- $1 \leq p \leq N, 1 \leq q \leq N, p \neq q$ 을 만족해야 한다.
- query 함수는 아래와 같이 동작한다.
 1. 모든 $0 \leq i < M$ 에 대해, U_i 번 정점과 V_i 번 정점을 연결하는 양방향 간선을 그래프에 추가한다.
 2. p 번 정점과 q 번 정점이 연결되어 있는지 판별한다.
 3. 1.에서 추가한 간선을 모두 그래프에서 제거한다.
 4. 2.에서 두 정점 p 와 q 가 연결되어 있었다면 1, 아니면 0을 반환한다.
- 여러분은 query 함수를 테스트 케이스마다 최대 13회 호출할 수 있다.

여러분은 아래와 같은 함수를 호출해야 한다.

```
void answer(int a, int b)
```

- a, b 의 쌍은 문제에 주어진 두 정수 A, B 의 쌍과 같아야 한다. 순서는 중요하지 않다.
- `answer` 함수는 테스트 케이스마다 정확히 한 번 호출되어야 한다.

제공되는 Sample Grader는 실제 채점에 활용되는 Grader와 다를 수 있음에 유의하라.

입력 형식

Sample Grader는 다음과 같은 정보를 표준 입력을 통하여 읽어들인다. 여러분은 어떠한 입력도 받으면 안된다.

첫번째 줄에 테스트 케이스의 총 개수를 의미하는 자연수 T 가 주어진다.
두번째 줄부터 T 개의 줄에 걸쳐, T 개의 테스트 케이스에 관한 정보가 주어진다.
($i + 1$)번째 줄에는 세 개의 자연수 N, A, B 가 사이에 공백을 두고 주어진다($1 \leq i \leq T$).

출력 형식

Sample Grader는 다음과 같은 정보를 표준 출력을 통하여 출력한다. 여러분은 어떠한 출력도 하면 안된다.

Sample Grader는 첫번째 줄부터 T 개의 줄에 걸쳐, 각 줄마다 여러분이 찾은 A, B 값을 출력한다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq T \leq 500$
- $2 \leq N \leq 100$
- $1 \leq A \leq N$
- $1 \leq B \leq N$
- $A \neq B$

입력과 출력의 예

standard input	standard output
2	2 3
4 3 2	1 2
5 1 2	

부연 설명

다음은 위의 입출력 예제에 대해서, Sample Grader와 Interaction하는 과정을 나타낸 것이다.

여러분이 작성한 코드	Grader	설명
	$T = 2$	Grader에서 T 의 값을 읽어들인다.
	$N = 4,$ $A = 3, B = 2$	첫번째 테스트 케이스의 정보를 읽어들인다.
	find(4) 호출	find 함수는 매 테스트 케이스마다 단 한 번씩 호출된다.
query({1, 2}, {3, 4}, 1, 3) 호출		
	return 1	1번 정점과 3번 정점은 연결되어 있다.
answer(2, 3) 호출		여러분은 $A = 2, B = 3$ 라고 예상하였고 이는 실제와 순서만 다를 뿐 값은 일치한다.
find($N = 4$) 종료		
	$N = 5,$ $A = 1, B = 2$	두번째 테스트 케이스의 정보를 읽어들인다.
	find(5) 호출	
answer(1, 2) 호출		여러분은 $A = 1, B = 2$ 라고 예상하였고 이는 실제와 정확히 일치한다.
find($N = 5$) 종료		

위의 Interaction 과정은 실제 정해와 전혀 관련이 없으며, 단지 이해를 돕기 위해 만든 예시임에 유의하라.

이 페이지는 의도적으로 비워 두었습니다.

Problem K. 정기 모임

입력 파일: standard input
출력 파일: standard output
시간 제한: 2 seconds
메모리 제한: 512 mebibytes

정점이 N 개인 트리가 주어진다. 각 정점은 1번부터 N 번까지 차례대로 번호가 부여되어 있다.
 i 번째 간선은 A_i 번 정점과 B_i 번 정점을 연결하며, 가중치는 C_i 다($1 \leq i < N$).

트리에서 두 정점 사이의 거리는 그 둘을 잇는 최단경로 상의 간선의 가중치의 최댓값으로 정의한다.
단, 같은 두 정점 사이의 거리는 0으로 정의한다.

트리에 사는 사람들이 Q 개의 모임을 개최하려 한다.

i 번째 모임에는 S_i 이상 E_i 이하의 번호를 가진 정점에 사는 사람들이 참석한다.

각 모임은 트리 상의 어떤 정점 v 에서 이루어진다. 사람들이 모이는 정점 v 를 잘 정하여, 모임에 참석하는 각 사람이 정점 v 까지 이동하는 거리의 최댓값을 최소화하고자 한다.

이때 이 최솟값을 구하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 트리의 정점 개수를 의미하는 자연수 N 과 모임이 개최되는 횟수를 의미하는 자연수 Q 가 사이에 공백을 두고 주어진다.

두번째 줄부터 $(N - 1)$ 개의 줄에 걸쳐, 트리의 간선에 관한 정보가 주어진다.

$(i + 1)$ 번째 줄에는 세 개의 자연수 A_i , B_i , C_i 가 사이에 공백을 두고 주어진다($1 \leq i < N$). 이는 A_i 번 정점과 B_i 번 정점을 연결하는 가중치 C_i 의 간선이 존재함을 의미한다.

$(N + 1)$ 번째 줄부터 Q 개의 줄에 걸쳐, Q 개의 모임에 관한 정보가 주어진다.

$(N + i)$ 번째 줄에는 i 번째 모임을 나타내는 두 개의 자연수 S_i 와 E_i 가 사이에 공백을 두고 주어진다($1 \leq i \leq Q$).

출력 형식

첫번째 줄부터 Q 개의 줄에 걸쳐, 답을 차례대로 출력한다.

i 번째 줄에는 i 번째 모임에 대한 답을 출력한다($1 \leq i \leq Q$).

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $1 \leq N \leq 300,000$
- $1 \leq Q \leq 300,000$
- $1 \leq A_i \leq N$ ($1 \leq i < N$)
- $1 \leq B_i \leq N$ ($1 \leq i < N$)
- $A_i \neq B_i$ ($1 \leq i < N$)
- $1 \leq C_i \leq 10^9$ ($1 \leq i < N$)
- $1 \leq S_i \leq E_i \leq N$ ($1 \leq i \leq Q$)

입력과 출력의 예

standard input	standard output
3 3	1
1 2 1	2
2 3 2	2
1 2	
2 3	
1 3	

Problem L. 분수 계산

입력 파일: standard input
출력 파일: standard output
시간 제한: 1 second
메모리 제한: 512 mebibytes

0 이상 1 미만인 두 수 x, y 에 대하여, 그 둘의 거리 $d(x, y)$ 를 다음과 같이 정의하자.

$$d(x, y) = \min \{|x - y|, 1 - |x - y|\}$$

$0 \leq A_1 \leq A_2 \leq \dots \leq A_N < 1$ 을 만족하는 길이 N 의 수열 A 에 대하여, 이의 분포 정도 $f(A)$ 를 다음과 같이 정의하자.

$$f(A) = \sum_{1 \leq i < j \leq N} d(A_i, A_j)$$

길이 N 인 수열 A 가 주어질 때, 길이가 N 이면서 A 보다 분포 정도가 큰 수열이 존재하는지 판별하는 프로그램을 작성하시오.

즉, $f(A) < f(B)$ 를 만족하는 길이 N 의 수열 B 가 존재하는지 여부를 판별하는 프로그램을 작성하시오.

입력 형식

첫번째 줄에 수열의 길이를 의미하는 자연수 N 이 주어진다.

두번째 줄에 N 개의 유리수 $A_1, A_2, \dots, A_N$ 이 사이에 공백을 두고 주어진다.

하나의 유리수는 " p/q " 형태로 주어지며, 이는 그 수가 $\frac{p}{q}$ 임을 의미한다. 여기서 p 와 q 는 정수며,

$0 \leq p < q \leq 10^9$ 임이 보장된다.

출력 형식

만약 길이가 N 이면서 A 보다 분포 정도가 큰 수열이 존재한다면, 첫번째 줄에 "YES"를 출력한다.

만일 그렇지 않다면, 첫번째 줄에 "NO"를 출력한다.

제약 조건

모든 입력 데이터는 다음 조건을 만족한다.

- $2 \leq N \leq 10^6$
- $0 \leq A_i < 1$ ($1 \leq i \leq N$)
- $A_i \leq A_{i+1}$ ($1 \leq i < N$)

입력과 출력의 예

standard input	standard output
2 0/1 1/2	NO
2 0/1 2/3	YES
3 0/1 1/6 1/3	YES

이 페이지는 의도적으로 비워 두었습니다.