

Costruendo Grattacieli (skyscrapers)

Giorno	1
Lingua	Italiano
Limite di tempo:	3.5 secondi
Limite di memoria:	1024 megabyte

Stiamo per costruire una nuova città: Metropolis. La città verrà costruita in una griglia quadrata infinita e una volta completata sarà formata da n grattacieli, numerati da 1 a n . Ogni grattacielo occupa una cella differente della griglia. In ogni momento durante la costruzione, le celle che non contengono un grattacielo sono considerate vuote.

Ti vengono date le coordinate pianificate per gli n grattacieli. Il tuo compito è trovare un ordine in cui possono essere costruiti soddisfacendo le regole elencate sotto.

- L'impresa edile ha solo una gru, quindi Metropolis dovrà essere costruita un grattacielo alla volta.
- Il primo grattacielo può essere costruito ovunque nella griglia.
- Ogni grattacielo costruito dopo il primo dovrà condividere un lato o un angolo con almeno uno dei grattacieli costruiti precedentemente (in modo da riuscire ad allinearli alla griglia più facilmente).
- Durante la costruzione di un grattacielo ci deve essere una strada per portare il materiale al sito di costruzione dall'esterno della città muovendosi solo attraverso celle vuote che condividono un lato. In altre parole, ci deve essere un percorso di caselle nella griglia che condividono un lato in grado di collegare il grattacielo con una cella (r, c) con $|r| > 10^9$ e/o $|c| > 10^9$.

Se esiste una soluzione, denotiamo con s_1, \dots, s_n gli indici dei grattacieli nell'ordine in cui devono essere costruiti. Ci sono due tipi di subtask:

Tipo 1: Puoi produrre qualsiasi ordine valido.

Tipo 2: Devi trovare l'ordine che massimizza s_n . Tra questi, devi trovare quello che massimizza s_{n-1} . E così via. In altre parole devi trovare l'ordine di costruzione valido $(s_n, s_{n-1}, \dots, s_1)$ che sia lessicograficamente massimo.

File di input

La prima riga contiene un singolo intero n ($1 \leq n \leq 150,000$) – il numero di grattacieli.

La seconda riga contiene un singolo intero t ($1 \leq t \leq 2$) che indica il tipo di subtask descritto sopra.

Ciascuna delle n successive righe i contiene due interi separati da spazi r_i e c_i ($|r_i|, |c_i| \leq 10^9$) che denotano le coordinate della cella contenente il grattacielo i .

È garantito che non esistono due grattacieli su una stessa cella.

File di output

Se è impossibile costruire i grattacieli secondo le regole date, stampa una singola riga contenente la stringa "NO".

Altrimenti stampa $n + 1$ righe. La prima di queste righe dovrà contenere la stringa "YES". Per ogni i , la i -esima delle righe rimanenti dovrà contenere un singolo intero s_i .

Nel subtask con $t = 1$, se esistono più ordini validi, puoi stampare uno qualsiasi di esso.

Assegnazione del punteggio

Subtask 1 (8 punti): $t = 1$ e $n \leq 10$

Subtask 2 (14 punti): $t = 1$ e $n \leq 200$

Subtask 3 (12 punti): $t = 1$ e $n \leq 2,000$

Subtask 4 (17 punti): $t = 2$ e $n \leq 2,000$



Subtask 5 (20 punti): $t = 1$
Subtask 6 (10 punti): $t = 2$, $n \leq 70,000$ e $|r_i|, |c_i| \leq 900$ per ogni i
Subtask 7 (19 punti): $t = 2$

Esempi

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

Note

Nel primo esempio ci sono tre grattacieli in riga.
Tutti questi possono sempre essere raggiunti da fuori Metropolis, e ci sono quattro ordini di costruzione che preservano la connettività:

- 1, 2, 3
- 2, 1, 3
- 2, 3, 1
- 3, 2, 1

Dato che $t = 2$ dobbiamo scegliere la prima opzione.
Nel secondo esempio l'unica differenza dal primo è che il grattacielo 2 condivide solo angoli con i grattacieli 1 e 3, il medesimo insieme di ordini di costruzione del primo esempio è valido. Dato chapter $t = 1$ ognuno di questi è corretto.
Nel terzo esempio Metropolis è disconnessa: ovviamente non possiamo costruirla.