

Assegnazione del punteggio

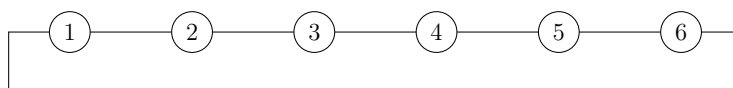
Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [5 punti]:** Caso d'esempio
- **Subtask 2 [7 punti]:** I nodi di rete sono collegati in sequenza, come in figura



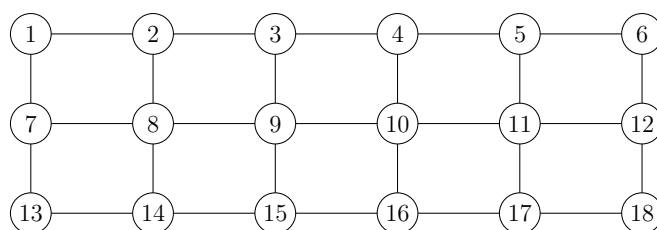
È sempre garantito che il primo nodo della sequenza è il nodo 1, e che il nodo k segue sempre il nodo $k - 1$, per ogni $k \geq 2$.

- **Subtask 3 [10 punti]:** I nodi di rete sono collegati ad anello, come in figura



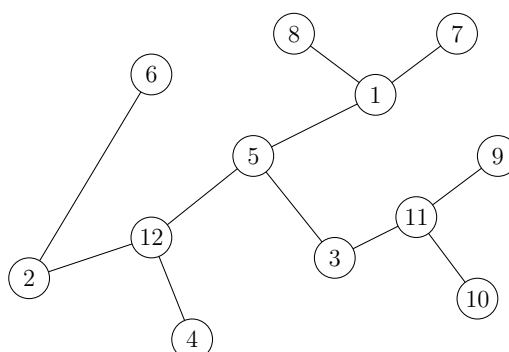
È sempre garantito che il nodo N è collegato al nodo 1, e che il nodo k segue il nodo $k - 1$, per ogni $2 \leq k \leq N$.

- **Subtask 4 [11 punti]:** I nodi di rete sono collegati a griglia, come in figura



È sempre garantito che i nodi seguiranno una numerazione per righe.

- **Subtask 5 [17 punti]:** Tra ogni coppia di nodi di rete esiste un unico percorso che li collega, come in figura



- **Subtask 6 [22 punti]:** $N, M \leq 1000$.
- **Subtask 7 [28 punti]:** Nessuna limitazione specifica (vedi la sezione **Assunzioni**).

Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp` o `.pas`.

👉 Tra gli allegati a questo task troverai un template (`bottleneck.c`, `bottleneck.cpp`, `bottleneck.pas`) con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C/C++	<pre>int Analizza(int N, int M, int W, int L, int arco_da[], int arco_a[], int capacita[], int R, int C);</pre>
Pascal	<pre>function Analizza(N, M, W, L: longint; var arco_da, arco_a, capacita: array of longint; R, C: longint): longint;</pre>

dove:

- N rappresenta il numero di nodi di rete.
- M rappresenta il numero di collegamenti.
- W e L sono rispettivamente il computer di William e quello di Luca.
- `arco_da` e `arco_a` sono due array di dimensione M che rappresentano i collegamenti. L' i -esimo collegamento di rete connette (in modo bidirezionale) i nodi `arco_da[i]` e `arco_a[i]`. È garantito che lo stesso collegamento non venga mai ripetuto.
- `capacita` è un array di dimensione M . L'intero `capacita[i]` rappresenta la capacità dell' i -esimo collegamento, in megabit al secondo.
- R e C sono parametri speciali che di norma valgono -1 . L'unica eccezione è il caso della topologia a griglia (vedi **Subtask 4**), in cui R e C rappresentano rispettivamente il numero di righe e di colonne della griglia.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati di input dal file `input.txt`, a quel punto chiama la funzione `Analizza` che dovete implementare. Il grader scrive sul file `output.txt` la risposta fornita dalla funzione `Analizza`.

Nel caso vogliate generare un input per un test di valutazione, il file `input.txt` deve avere questo formato:

- Riga 1: contiene l'intero N , che rappresenta il numero di nodi di rete, l'intero M , che rappresenta il numero di collegamenti, gli interi W e L , che sono i computer di William e di Luca rispettivamente, gli interi R e C , che di norma valgono -1 e nel caso della topologia a griglia rappresentano rispettivamente il numero di righe e di colonne della griglia.
- Righe 2, ..., $M + 1$: l' i -esima riga contiene tre interi x_i , y_i , z_i con $1 \leq x_i \leq N$ e $1 \leq y_i \leq N$, che rappresentano un collegamento tra i nodi x_i e y_i con capacità z_i .

Il file `output.txt` invece ha questo formato:

- Riga 1: contiene il valore restituito dalla funzione `Analizza`.

Assunzioni

- $2 \leq N \leq 100\,000$.
- $1 \leq M \leq 1\,000\,000$.
- Le capacità dei collegamenti sono interi compresi tra 0 e 1 000 000 000.
- Quando R e C non valgono entrambi -1 , vale che $1 \leq R, C \leq 300$.
- Il grafo della rete è connesso, cioè tutti ogni nodo è raggiungibile da tutti gli altri.
- Nessun collegamento connette un nodo con se stesso.
- Per ogni coppia di nodi c'è al più un collegamento che li connette.

Esempi di input/output

input.txt	output.txt
12 17 2 8 3 4 1 2 1 2 3 8 3 4 3 1 5 3 2 6 7 3 7 2 4 8 9 5 6 4 5 9 6 6 7 5 7 8 6 9 10 1 10 11 2 11 12 5 10 6 8 11 7 5 8 12 2	2

Questo caso di input corrisponde all'esempio spiegato nel testo.