

Sfortuna

Lo sfortunatissimo Re di *Molto, Molto, Molto Lontano* è talmente sfortunato da aver attirato sul suo Regno una sciagurata pioggia di meteoriti. Tutte le strade sono state distrutte e, come se non bastasse, si sta avvicinando l'inizio dei *Magici Giochi Automobilistici*, il quinquennale evento automobilistico che proprio quest'anno si tiene nel suddetto Regno.

Il Re è venuto a conoscenza del fatto che stai seguendo il corso di Algoritmi e Strutture Dati presso l'Università degli Studi di Ferrara. Per evitare una tremenda figuraccia, ti implora di sillabare una potente *magia algoritmica* in C in grado di computare quali strade conviene ricostruire per collegare tutte le città del Regno tra loro, spendendo meno monete d'oro possibile.

Hai una missione: portala a termine e le tue gesta verranno narrate fino alle remote locande del Regno di *Troppo, Troppo, Troppo Lontano*.

Input

L'input deve essere letto da un file "input.txt".

La prima riga dell'input consiste del numero di città N e di M proposte di strade fattibili da costruire prima dei *Magici Giochi Automobilistici*.

Seguono M righe, ognuna delle quali descrive una possibile strada con una tripla di interi u , v e w separati da uno spazio. La strada che può essere costruita tra le città u e v , può essere realizzata spendendo w monete d'oro. La numerazione delle città inizia da 0.

Output

L'output deve essere scritto su un file "output.txt".

L'output consiste nel più piccolo numero di monete d'oro che devono essere spese per collegare tra loro tutte le città. In particolare, considerate due città qualsiasi u e v del Regno, deve sempre esistere un qualche percorso $u \rightsquigarrow v$ che porta a v partendo da u .

Vincoli

- $1 \leq N \leq 10^4$;
- $1 \leq M \leq 10^5$;
- $1 \leq W_i \leq 2^{40}$;
- La soluzione è sicuramente $< 2^{63}$;
- **tempo limite:** 2 s;
- **memoria limite:** 512 MiB.

Punteggio e casistiche

Risolvere questo problema garantisce 3/30 punti in più all'esame scritto finale. Ricorda: i punti sono cumulativi tra i vari esercizi, ed è possibile accumulare un massimo di 8/30 punti.

Ogni soluzione è testata considerando un certo numero di input, di difficoltà incrementale:

- 1: tutti i vincoli (100 punti);

Esempio

Input (input.txt)	Output (output.txt)
7 9 0 1 7 1 2 21 0 2 14 0 3 30 3 2 10 2 4 1 4 5 6 4 6 9 5 6 4	42

Il caso d'esempio descrive il grafo seguente, dove in rosso sono indicati gli archi da scegliere per spendere meno monete d'oro possibili.

