

Busy Beavers

Un Alacre Castoro (o [Busy Beaver](#)) è una [Macchina di Turing](#) che, presa in input una stringa binaria di una certa lunghezza, produce in output una stringa con il maggior numero di 1 possibile senza sfruttare cicli infiniti. Attualmente (2024), l'Alacre Castoro più produttivo con input binario di lunghezza 6 riesce a stampare il numero 1 più di $10 \uparrow 15$ (riferimento [qui](#)) ovvero $10^{10 \dots 10}$ volte (in totale l'esponente 10 è applicato 15 volte).

L'Università degli Studi di Ferrara, in particolare il Dipartimento di Matematica e Informatica, è il luogo preferito dei Castori Algoritmisti, la famigerata banda ferrarese di castori-algoritmisti che trascorre il suo tempo studiando la particolare Macchina di Turing citata sopra e, dietro le quinte, organizza ogni anno il corso di Algoritmi e Strutture Dati del corso di Informatica.

Ultimamente, il Capo Castoro Odiugoc Civaics è molto impegnato: sta dimostrando che un certo Alacre Castoro che lavora con alfabeto senario anziché binario può produrre un numero di 1 pari a $10 \uparrow 10 \uparrow 10^{10^{15}}$ a partire da una stringa di lunghezza 2. Non avendo troppo tempo libero, decide di affidarti una missione.

Devi visitare tutte le case dei Castori Algoritmisti, situate nelle gallerie sotterranee del Dipartimento di Matematica e Informatica, e portare della legna in ciascuna casa. Per portare a termine il tuo compito, disponi di una vasta scelta di carretti dalle dimensioni differenti. Puoi modellare la rete in cui ti sposterai per consegnare la legna come un grafo non direzionato e pesato, in cui pesi più grandi indicano gallerie più larghe. Ogni carretto di larghezza w riesce a passare in ogni galleria di spessore maggiore o uguale a w .

In effetti, conosci la mappa dei sotterranei a memoria, e sai che purtroppo alcuni carretti sono troppo larghi per attraversare delle gallerie molto strette. Fortunatamente, disponi anche di un Rimpicciolitore, grazie al quale puoi superare una qualsiasi galleria con qualsiasi carretto: attento però, l'effetto del Rimpicciolitore svanisce subito dopo aver attraversato una singola galleria (ovvero, un arco nel grafo che modella la mappa).

Per ogni carretto i -esimo a tua disposizione, la cui larghezza è W_i , ti chiedi: immaginando di partire da una qualsiasi casa, qual è il numero più piccolo di utilizzi del Rimpicciolitore necessario per poter consegnare la legna ad ogni altra casa?

Input

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

La prima riga dell'input contiene tre interi: il numero di case N , il numero di gallerie K e il numero di carretti a disposizione T .

Le successive K righe contengono tre interi A_i, B_i, V_i per indicare che le case A_i e B_i sono connesse da una galleria di larghezza V_i . La numerazione delle case inizia da 0.

Le T righe finali contengono un intero W_i ciascuna, ovvero la larghezza di ogni carretto.

Output

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

L'output consiste di T righe, ciascuna contenente un singolo intero: il minimo numero di utilizzi del Rimpicciolitore per cui tutte le case possono essere raggiunte con il carretto.

Vincoli

- $1 \leq N \leq 50000$;
- $1 \leq K \leq 600000$;

- $1 \leq T \leq 100000$;
- $0 \leq W_i \leq 50000$, $i = 0 \dots (T - 1)$;
- **tempo limite:** 2 s;
- **memoria limite:** 64 MiB.

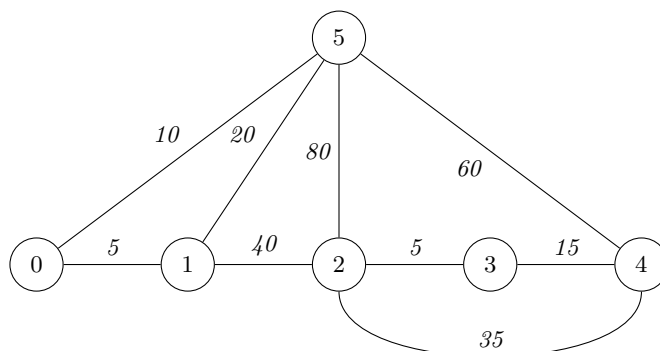
Punteggio e casistiche

Questo problema è affrontato durante il tutorato, e **non garantisce punti in più all'esame scritto finale**. Il problema è testato sull'unica seguente casistica:

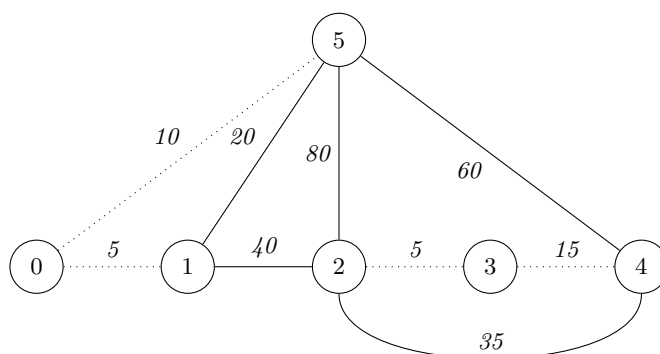
- 1: tutti i vincoli, e le gallerie sotterranee si modellano come un grafo connesso (80 punti).
- 2: tutti i vincoli, ma le gallerie sotterranee non sono modellate come un grafo connesso. Vale a dire che alcune case sono completamente irraggiungibili, indipendentemente dalla larghezza del carretto, a meno che non si utilizzi una volta il Rimpicciolitore (20 punti).

Esempi

Il primo caso d'esempio si può modellare a partire dal grafo di seguito.

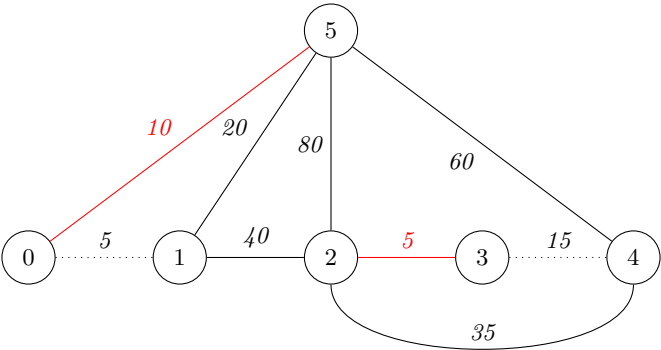


Consideriamo il primissimo carretto, con larghezza 20. Purtroppo, senza ricorrere mai al Rimpicciolitore, le case 0 e 3 non sono mai raggiungibili. La situazione è rappresentata di seguito.

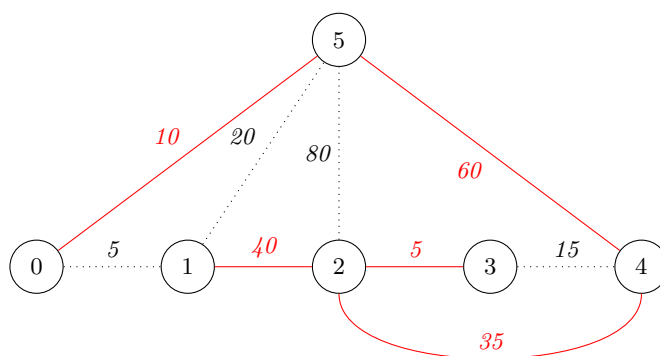


Per raggiungere tutte le case dei castori, è necessario utilizzare almeno 2 volte il Rimpicciolitore. Nel grafo che segue, i 2 utilizzi del Rimpicciolitore sono evidenziati in rosso (si noti che la scelta degli archi potrebbe non essere univoca).

Input (input.txt)	Output (output.txt)
6 9 5	2
0 1 5	5
0 5 10	0
1 2 40	3
1 5 20	2
2 3 5	
2 4 35	
2 5 80	
3 4 15	
4 5 60	
20	
90	
3	
60	
30	
2 1 5	0
0 1 3	0
1	0
2	1
3	1
4	
5	

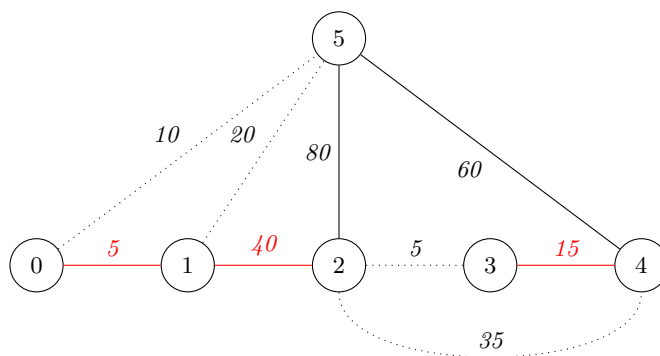


Nel caso del secondo carrello, con larghezza 90, il Rimpicciolitore deve essere utilizzato almeno una volta per ogni coppia di nodi, e la risposta è 5. Ad esempio, è possibile scegliere di utilizzare il Rimpicciolitore nei seguenti archi:



Il terzo carretto, con larghezza 3, è abbastanza stretto da poter passare ovunque: non utilizzo mai il Rimpicciolitore e dunque la risposta è 0.

Per il quarto carretto, con larghezza 60, la risposta è 3:



Per il quinto carretto, con larghezza 30, la risposta è 2:

