

Problema: CLO

Cloud computing

CEOI 2018, giorno 1. Memoria disponibile: 256 MB.

14.08.2018

Johnny sta finanziando Bytecomp, un'azienda che offre potenza di calcolo nel cloud. Questo genere di aziende solitamente hanno molti computer nei quali vengono eseguiti i calcoli dei loro clienti.

Johnny non ha ancora comprato alcun server. Lui è appena andato nel negozio di computer e si è fatto dare la lista degli n computer disponibili. Ogni computer è caratterizzato dal numero di core c_i , la frequenza di clock f_i e il prezzo v_i . Ciascuno di questi computer ha c_i core che non interferiscono tra loro, quindi possono essere assegnati a computazioni differenti.

Quando un cliente effettua un ordine di risorse egli specifica il numero di core C_j richiesti, e la frequenza minima di clock F_j che ciascuno di questi deve avere. Un ordine contiene anche il prezzo V_j che il cliente pagherà nel caso l'ordine venga accettato. Se un ordine viene accettato, Bytecomp deve garantire accesso esclusivo ai core che vengono assegnati al cliente. Johnny deve scegliere C_j core (anche da macchine diverse), ognuno con frequenza almeno pari a F_j . Questi core non possono essere usati da altri ordini.

Aiuta Johnny a guadagnare il più possibile: scegli il sottoinsieme ottimo di ordini da accettare e il sottoinsieme ottimo di computer da comprare al negozio tali che siano in grado di soddisfare tutti gli ordini accettati. Il tuo obiettivo è di massimizzare il profitto, cioè la differenza tra i guagni ottenuti e la spesa per i computer.

Input

La prima riga dello standard input contiene un intero n ($1 \leq n \leq 2000$), il numero di computer disponibili al negozio. Ognuna delle successive n righe descrive un computer. Queste consistono in 3 interi separati da spazi c_i , f_i e v_i ($1 \leq c_i \leq 50$, $1 \leq f_i \leq 10^9$, $1 \leq v_i \leq 10^9$) che rappresentano il numero di core, la frequenza e il prezzo, rispettivamente.

La riga successiva contiene un intero m ($1 \leq m \leq 2000$), il numero di ordini. Ognuna delle successive m righe descrive un ordine. Queste consistono in 3 interi separati da spazi C_j , F_j e V_j ($1 \leq C_j \leq 50$, $1 \leq F_j \leq 10^9$, $1 \leq V_j \leq 10^9$) che rappresentano il numero di core richiesti, la frequenza di clock minima e il budget del cliente, rispettivamente.

Output

L'unica riga dello standard output deve contenere un intero, il profitto totale massimo che può essere ottenuto.

Grading

L'insieme dei test è diviso nei seguenti subtask con limitazioni aggiuntive. I test in ogni subtask consistono in uno o più gruppi di test. Ogni gruppo di test contiene uno o più test case.

Subtask	Limitazioni	Punti
1	$n \leq 15$	18
2	$m \leq 15$	18
3	$n, m \leq 250$, $c_i = C_j = 1$	18
4	$f_i = F_j = 1$	18
5	$v_i = V_j = 1$	18
6	nessuna limitazione aggiuntiva	10

Esempio

Per l'input:

```
4
4 2200 700
2 1800 10
20 2550 9999
4 2000 750
3
1 1500 300
6 1900 1500
3 2400 4550
```

l'output corretto è:

```
350
```

Spiegazione dell'esempio: Ci sono 4 computer e 3 ordini. La scelta ottima consiste nel comprare i 2 quad-core che costano 700 e 750 (1450 in totale) e accettare i primi due ordini per guadagnare $300 + 1500 = 1800$ in totale. Quindi abbiamo 4 core con frequenza 2000 e altri 4 con frequenza 2200. Possiamo assegnarne 6 qualunque per il secondo ordine (tutti hanno almeno 1900 di frequenza) e uno dei 2 rimasti per il primo ordine (hanno entrambi almeno 1500 di frequenza). Un core quindi non verrà quindi usato, ciò è permesso.

Il profitto totale è di $1800 - 1450 = 350$.