

### Spaghetticode

Thomas und Matthias haben beschlossen, am diesjährigen **Ludum Dare** teilzunehmen. Da sie letztes Jahr Matthias seine Game-Engine verwendet haben, wollen sie diesmal Thomas seine Game-Engine verwenden. Das führt natürlich (wie hätte es auch anders sein können?) wieder zu Problemen.



Aber zuerst zu ihrem Spiel. Du bist in einem Labyrinth gefangen. Dieses besteht aus  $n$  Räumen und  $m$  Gängen. Für jeden Gang ist eine Länge  $l$  sowie eine Geschwindigkeit  $v$  definiert. Die Geschwindigkeit gibt an, wie schnell du dich vom Anfang des Ganges bis zum Ende bewegst. Außerdem kann jeder Gang nur in einer Richtung begangen werden.

Nun zu den vorher erwähnten Problemen mit der Game-Engine. Leider wirkt der Physics-Teil der Engine wie aus den 80er Jahren und ist somit komplett veraltet. Einerseits ist aufgrund der veralteten Technik, die Beschleunigung zwischen dem Ende des einen Ganges und dem Anfang des nächsten Ganges sofortig. Andererseits stürzt die Game-Engine komplett ab, wenn die Geschwindigkeit eines Ganges undefiniert ist (das wird durch  $v = 0$  repräsentiert). Das bedeutet, sollte für einen Gang  $v = 0$  sein, setzt die Game-Engine (aus purer Verzweiflung)  $v$  auf die Geschwindigkeit des letzten begangenen Ganges.

Da nur noch wenig Zeit bis zum Ende des Ludum Dare bleibt, können Thomas und Matthias die Game-Engine nicht mehr fixen und müssen das Spiel in diesem Zustand herausbringen.

Sie bitten dich jetzt zumindest für ein gegebenes Labyrinth herauszufinden, was der schnellste Weg vom Startraum zum Zielraum  $n - 1$  ist. (Natürlich mit den ganzen Game-Engine Problemen)

### Anmerkungen

- Die Zeit  $t$  um durch einen Gang zu kommen ergibt sich aus  $t = l/v$   
Es ist daher notwendig einen „double“ im code zu verwenden. (Gleitkommazahlen)
- Du kannst davon ausgehen, das es keine Zeit braucht um einen Raum zu durchqueren.  
Es zählt lediglich die Summe der Zeiten der besuchten Gänge.
- Des weiteren gibt es zwischen zwei benachbarten Räumen  $a$  und  $b$  höchstens einen Gang.
- Am Anfang bist du in Raum 0 und deine momentane Geschwindigkeit ist 70.
- Es gibt immer einen Weg vom Startraum zum Zielraum.

### Eingabe

Die erste Zeile besteht aus zwei Zahlen  $n, m$ .

Die Anzahl der Räume und Gänge. Der Zielraum ist  $d = n - 1$ .

Dannach kommen  $m$  Zeilen, welche die Gänge repräsentieren. Jede Zeile besteht aus vier Zahlen  $a, b, v$  und  $l$ .

- $a$  und  $b$  geben den Anfang des Ganges sowie das Ende an. (Nicht vergessen, er kann nur durch diese Richtung begangen werden!)
- $v$  die Geschwindigkeit, die du in diesem Gang gehst. Wenn  $v = 0$  ist bedeutet dies, dass  $v =$  das  $v$  des letzten Ganges.
- $l$  gibt die Länge des Ganges an.

## Ausgabe

Gib in der ersten Zeile die Zeit aus, die du brauchst, um auf dem optimalen Weg, zum Zielraum zu kommen (eine Dezimalzahl<sup>1</sup>). Dannach gib die Reihenfolge der Knoten aus die du auf dem schnellsten Weg nach  $n - 1$  besuchst. Beginnend bei 0 bis  $n - 1$  (inklusive).

## Beispiel

Eingabe	Ausgabe
3 3 0 1 10 10 1 2 0 10 0 2 5 20	2.0 0 1 2

## Subtasks

Allgemein gilt:

- $1 \leq n \leq 10^3$
- $0 \leq m \leq 10^4$
- $0 \leq v \leq 500$
- $1 \leq l \leq 10^7$

**Subtask 1 (15 Punkte):** Alle  $v \neq 0$

**Subtask 2 (85 Punkte):** Keine weiteren Einschränkungen

Du bekommst 15% der Punkte wenn du nur die benötigte Zeit ausgibst.

---

<sup>1</sup>Bei `printf` als Format `%lf` verwenden.