2. 宝藏

（treasure.cpp/c/pas）

【问题描述】

参与考古挖掘的小明得到了一份藏宝图，藏宝图上标出了 n 个深埋在地下的宝藏屋，也给出了这 n 个宝藏屋之间可供开发的 m 条道路和它们的长度。

小明决心亲自前往挖掘所有宝藏屋中的宝藏。但是，每个宝藏屋距离地面都很远，也就是说，从地面打通一条到某个宝藏屋的道路是很困难的，而开发宝藏屋之间的道路则相对容易很多。

小明的决心感动了考古挖掘的赞助商，赞助商决定免费赞助他打通一条从地面到某个宝藏屋的通道，通往哪个宝藏屋则由小明来决定。

在此基础上，小明还需要考虑如何开凿宝藏屋之间的道路。已经开凿出的道路可以任意通行不消耗代价。每开凿出一条新道路，小明就会与考古队一起挖掘出由该条道路所能到达的宝藏屋的宝藏。另外，小明不想开发无用道路，即两个已经被挖掘过的宝藏屋之间的道路无需再开发。

新开发一条道路的代价是：

这条道路的长度 × 从赞助商帮你打通的宝藏屋到这条道路起点的宝藏屋所经过的宝藏屋的数量（包括赞助商帮你打通的宝藏屋和这条道路起点的宝藏屋）。

请你编写程序为小明选定由赞助商打通的宝藏屋和之后开凿的道路，使得工程总代价最小，并输出这个最小值。

【输入格式】

输入文件名为 treasure.in。

第一行两个用空格分离的正整数 n 和 m，代表宝藏屋的个数和道路数。

接下来 m 行，每行三个用空格分离的正整数，分别是由一条道路连接的两个宝藏屋的编号（编号为 1~n），和这条道路的长度 v。

【输出格式】

输出文件名为 treasure.out。

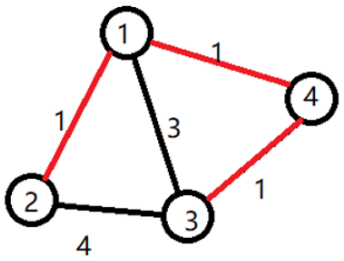
输出共一行，一个正整数，表示最小的总代价。

【输入输出样例 1】

treasure.in	treasure.out
4 5 1 2 1 1 3 3 1 4 1 2 3 4 3 4 1	4

见选手目录下的 treasure/treasure1.in 与 treasure/treasure1.ans

【输入输出样例 1 说明】



小明选定让赞助商打通了 1 号宝藏屋。小明开发了道路 1→2，挖掘了 2 号宝藏。开发了道路 1→4，挖掘了 4 号宝藏。还开发了道路 4→3，挖掘了 3 号宝藏。工程总代价为：1×1 + 1×1 + 1×2 =4

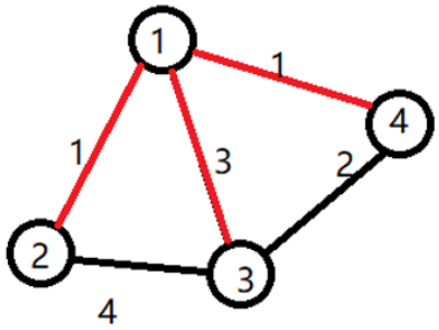
(1→2) (1→4) (4→3)

【样例输入输出 2】

treasure.in	treasure.out
4 5 1 2 1 1 3 3 1 4 1 2 3 4 3 4 2	5

见选手目录下的 treasure/treasure2.in 与 treasure/treasure2.ans。

【输入输出样例 2 说明】



小明选定让赞助商打通了 1 号宝藏屋。小明开发了道路 1→2，挖掘了 2 号宝藏。开发了道路 1→3，挖掘了 3 号宝藏。还开发了道路 1→4，挖掘了 4 号宝藏。工程总代价为：1×1 + 3×1 + 1×1 =5

(1→2) (1→3) (1→4)

【输入输出样例 3】

见选手目录下的 treasure/treasure3.in 和 treasure/treasure3.out。

【数据规模与约定】

对于 20%的数据：

保证输入是一棵树，1≤n≤8，v≤5000 且所有的 v 都相等。

对于 40%的数据：

1≤n≤8，0≤m≤1000，v≤5000 且所有的 v 都相等。

对于 70%的数据：

1≤n≤8，0≤m≤1000，v≤ 5000

对于 100%的数据：

1≤n≤12，0≤m≤1000，v≤ 500000