Contest 12

1、素数（prime）

1s/256M

【问题描述】

我们定义v(i)表示小于等于i的最大素数，u(i)表示大于i的最小素数，求

的值，以最简分数的形式输出。

【输入描述】

第一行一个正整数t，表示测试数据数量；

每组测试数据一行一个正整数n。

【输出描述】

对于每组测试数据输出一个答案。

【样例】

|  |  |
| --- | --- |
| Sample Input | Sample Output |
| 2 2 3 | 1/6 7/30 |

【数据范围】

对于30%的数据，t <= 10,n <= 10000

对于100%的数据，t <= 500,n <= 1e9

2、精灵（go）

1s/256M

【问题描述】

Branimirko是一个对可爱精灵宝贝十分痴迷的玩家。最近，他闲得没事组织了一场捉精灵的游戏。游戏在一条街道上举行，街道上一侧有一排房子，从左到右房子标号由1到n。

刚开始玩家在k号房子前。有m个精灵，第i只精灵在第A[i]栋房子前，分值是B[i]，以及它在T[i]秒内（含）存在，之后消失。Branimirko可以选择移动至相邻的房子，耗时1秒。抓住精灵不需要时间，精灵被抓住后消失。时间从第1秒开始。Branimirko能最多获得多少分值和。

【输入描述】

输入的第1行为三个正整数n，k，m。

接下来m行描述精灵的信息，分别为A[i]，B[i]，T[i]。

【输出描述】

输出Branimirko能最多获得多少分值和。

【样例】

|  |  |  |
| --- | --- | --- |
|  | go.in | go.out |
| 样例1 | 10 5 4  1 30 4  3 5 7  7 10 12  9 100 23 | 115 |

【样例解释】

很遗憾，它恰好不能抓住在一号房子前的精灵。

如果T[1]改成5，答案就是145。

【数据范围】

20%的数据：

40%的数据：

对于100%的数据：，所有数为正整数。

3、秘密通道（portal）

1s/256M

【问题描述】

有一副n\*m的地图，有n\*m块地，每块是下列四种中的一种：

墙：用#表示，墙有4个面，分别是前面，后面，左面，右面。

起点：用C表示，为主角的起点，是一片空地。

终点：用F表示，为主角的目的地，是一片空地。

空地：用 . 表示。

其中除了墙不能穿过，其他地方都能走。

主角有以下3种操作：

1.移动到相邻的前后左右的地方，花费一个单位时间。

2.向前后左右其中一个方向发射子弹，子弹沿直线穿过，打在最近的一堵墙的一面，然后墙的这面就会形成一个开口通往秘密通道。同一时间最多只能有两个开口，若出现有3个开口，出现时间最早的开口会立即消失。该操作不用时间。

3.可以从一个与开口相邻的空地跳进去，进入秘密通道，从另外一个开口正对的空地跳出来。这个过程花费一个单位时间。

地图四周都是墙，问主角最少用多少时间从C走到F。C和F只会出现一次。

【输入描述】

第一行输入两个正整数n，m。

接下来n行，每行m个字符描述地图。

【输出描述】

输出1个整数，表示最短时间完成路途。如果无解输出nemoguce。

【样例】

|  |  |  |
| --- | --- | --- |
|  | portal.in | portal.out |
| 样例1 | 4 4  ####  #.F#  #C.#  #### | 2 |
| 样例2 | 6 8  ########  #.##..F#  #C.##..#  #..#...#  #.....##  ######## | 4 |
| 样例3 | 4 5  #####  #C#.#  ###F#  ##### | nemoguce |

【样例解释】

总共用到8次操作，时间之和为4。如下图所示

1.向左射一枪，在(3,1)的右面出现开口。

2.向下射一枪，在(6,2)的上面出现开口。

3.向左从(3,1)进入秘密通道，从(6,2)中出来，到达(5,2)。用1单位时间。

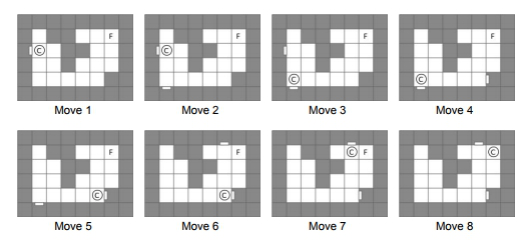
4.向右射一枪，在(5,7)的左面出现开口，(3,1)右面的开口消失。

5.走进(6,2)的开口，出来到(5,6)。用1单位时间。

6.向上射一枪，在(1,6)的下面出现开口。

7.经过秘密通道，走到(2,6)。用1单位时间。

8.走到终点。用1单位时间。



【数据范围】

对于50%的数据，。

对于100%的数据。

4、小x游世界树（yggdrasil）

1s/256M

【问题描述】

小x得到了一个(不可靠的)小道消息，传说中的神岛阿瓦隆在格陵兰海的某处,据说那里埋藏着亚瑟王的宝藏，这引起了小x的好奇，但当他想前往阿瓦隆时发现那里只有圣诞节时才能到达，然而现在已经春天了，不甘心的他将自己的目的地改成了世界树，他耗费了大量的时间，终于将自己传送到了世界树下。世界树是一棵非常巨大的树，它有着许许多多的枝条以及节点，每个节点上都有一个平台。好不容易来到传说中的世界树下，小x当然要爬上去看看风景。小x每经过一条边都会耗费体力值。然而世界树之主想给他弄（gáo）些（dǐan）麻（shì）烦（qíng），于是他在每条边上都设了一个魔法阵，当小x踏上那条边时会被传送回根节点，魔法阵只生效一次。这岂不是要累死小x？幸运的是，每个平台上都有无数个加速器，这些加速器可以让小x在当前节点所连的边上耗费的体力值减少，不同平台的加速器性能不一定相同，但同一个平台的加速器性能绝对相同。世界树之主给了小x一次“换根”的机会，他可以将世界树的任何一个节点变为根，但所有的边都不能改变。小x想问你，将根换为哪个节点能使小x爬到世界树上的每个节点耗费的体力值和最少。默认编号为1的点为初始根。

【输入描述】

第一行一个数n，表示有n个节点。

第二行n个数ai，表示每个平台上的加速器的性能。

第三至n+1行，每行三个数bi,ci,di分别表示这条无向边的起点，终点与耗费的能量值。

【输出描述】

第一行一个数，表示要换成的节点，如果有多个点为根时耗费的体力值都最小，则输出编号最小的那个。如果保持为1是最优的，就输出1。

第二行一个数，表示最小耗费的体力值。

【样例】

|  |  |  |
| --- | --- | --- |
|  | yggdrasil.in | yggdrasil.out |
| 样例1 | 4  2 1 3 3  1 2 3  1 3 4  2 4 6 | 1  9 |

【样例解释】

如果以第一个点为根，则需要耗费0(到1)+1(到2)+2(到3)+6(到4)=9的能量值。

如果以第二个点为根，则需要耗费2(到1)+0(到2)+4(到3)+5(到4)=11的能量值。

如果以第三个点为根，则需要耗费1(到1)+2(到2)+0(到3)+7(到4)=10的能量值。

如果以第四个点为根，则需要耗费5(到1)+3(到2)+7(到3)+0(到4)=15的能量值。

很明显以第一个点为根是最优的。

【数据范围】

对于20%的数据:n<=100

对于40%的数据:n<=1000

对于60%的数据:n<=8000

对于80%的数据:n<=100000

对于100%的数据:0<n<=700000;ai<=1000;1<=bi,ci<=n;di<=1000。

数据保证一个点的加速器性能绝对小于等于它的所有的边所耗费的能量，保证所有节点都可以到达,保证没有数据与样例相同。