

华师一校内测试题

2022 年 8 月 12 日

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	二战狙击手	打砖块	投资	仓库建设
英文题目名称	sniper	brike	money	storage
可执行文件名	sniper	brike	money	storage
输入文件名	sniper.in	brike.in	money.in	storage.in
输出文件名	sniper.out	brike.out	money.out	storage.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
内存限制	256M	256M	256M	256M
题目类型	传统	传统	传统	传统

二、提交源代码文件名

对于C语言	sniper.c	brike.c	money.c	storage.c
对于C++语言	sniper.cpp	brike.cpp	money.cpp	storage.cpp

三、编译命令 (不包含任何优化开关)

对于C语言	gcc -o sniper sniper.c -lm	gcc -o brike brike.c -lm	gcc -o money money.c -lm	gcc -o storage storage.c -lm
对于C++语言	g++ -o sniper sniper.cpp -lm	g++ -o brike brike.cpp -lm	g++ -o money money.cpp -lm	g++ -o storage storage.cpp -lm

四、注意事项

- 每位选手提交一个以自己编号命名的文件夹，其中包含4个子文件夹，名称分别为：sniper、brike、money、storage，每个题目需要上交1个相应的源程序到对应的子文件夹中。多交1个文件夹或文件扣10分，直至考试成绩到0分为止。文件(夹)名称错误不得分。
- 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。
- C/C++中函数main()的返回值类型必须是int，程序正常结束时的返回值必须是0。
- 统一评测时采用的机器配置为:windows下lemon评测和全国评测系统下评测。
- 最终测试时，所有编译命令均不打开任何优化开关。
- 请尽力优化，会收获更多的部分得分。

二战狙击手 （sniper）

-----ytj

问题描述：

Jackson 是一名美国神射手，他凭着一只从德军缴获的毛瑟 kar98 步枪把德军搅得鸡犬不宁。这天，他来到了一座塔楼，埋伏在这里进攻德军。

首先，目标点都被编了号，我们认为德军只会在目标点出现（所有编号相邻的目标点在物理上也是相邻的）。这样做起了保密的作用，因为即便德军截获这张表，但由于不知道编号，看不懂，只能作罢。

其次，德军可能出现的每种军衔的人都被他赋予了一定的价值。如下面这张表所示：

价值	军衔 / 编号
100	帝国元帅 1
100	元帅 2
97	上将 3
96	中将 4
95	少将 5
90	准将 6
85	上校 7
80	中校 8
82	少校 9
78	上尉 10
75	中尉 11
70	少尉 12
65	军士长 13
55	技术军士 14
50	狙击手 15
15	上士 16
12	中士 17
10	一级下士 18
8	下士 19
5	代理下士 20
2	二等兵 21

由于美国具有先进的密码破译手段，所以 Jackson 得到了一张德军出现时间的列表（宿命论？）。在这张列表上，每行有三个数字 W_i 、 U_i 、 T_i 。其中 W_i 表示这个人的军衔（如上表所示编号）， U_i 表示这人出现位置的编号， T_i 则对应此人出现的时间（单位：分钟）。

同时，还有以下几个事实（呼呵?!）：

- 1) 虽然 Jackson 的狙击枪一开始（即定义为 0 时刻）可以对准任何位置，但是其后每分钟只能移动到编号相差不超过 2 的位置上（移动幅度过大会被敌人发现）。
- 2) 当敌人出现在 Jackson 的狙击镜中 Jackson 就可以将他射杀。（废话……）
- 3) 每次开枪后都必须要一分钟的时间来拉枪栓，也就是说，在开枪后的一分钟内 Jackson 什么都不能干（包括不能移动狙击步枪的瞄准位置）。
- 4) 所有出现的狙击手都必须被射杀（否则会死得很惨，对方用的是 AWP，而且开了作弊器）。输入数据保证这一点能做到。
- 5) Jackson 每一枪都能且只能击中一个敌人。
- 6) 不会有两个人在同时同地出现。

求：在满足上述规则的前提下，Jackson 能获得的最大价值总和是多少？

输入说明

第一行：	一个整数 N 表示敌人人数。（ $N \leq 8000$ ）
第二行..m+1 行：	每行有三个数字 W_i 、 U_i 、 T_i ，描述一个敌人。（ $U_i \leq 20$ 、 $T_i \leq 25000$ ）

Ps：输入数据保证按照敌人出现的先后排好序了。

输出说明：

单独一行表示可以获得的最大价值。

输入样例：

```
5
9 1 1
8 1 2
15 3 4
2 2 4
21 6 8
```

输出：

```
134
```

样例说明：

射击第一个出现的少校，第一个狙击手，第一个二等兵。（也就是被红色标记的人）

数据范围：

对 20% 的数据， $n \leq 100$

对 100% 的数据， $n \leq 8000$

其余数据范围在题面中有提及。

打砖块 (briker)

问题描述：

在一个凹槽中放置了 n 层砖块，最上面的一层有 n 块砖，第二层有 $n-1$ 块，……，最下面一层仅有一块砖。第 i 层的砖块从左至右编号为 $1, 2, \dots, i$ ，第 i 层的第 j 块砖有一个价值 $a[i,j]$ ($a[i,j] \leq 50$)。下面是一个有 5 层砖块的例子：

14	15	4	3	23
33	33	76	2	
2	13	11		
22	23			
31				

如果要敲掉第 i 层的第 j 块砖的话，若 $i=1$ ，可以直接敲掉它，若 $i>1$ ，则必须先敲掉第 $i-1$ 层的第 j 和第 $j+1$ 块砖。

你的任务是从一个有 n ($n \leq 50$) 层的砖块堆中，敲掉 ($m \leq 500$) 块砖，使得被敲掉的这些砖块的价值总和最大。

【输入格式】

从文件 `briker.in` 中读入数据，数据的第一行为两个正整数，分别表示 n, m ，接来的第 i 每行有 $n-i+1$ 个数据，分别表示 $a[i,1], a[i,2], \dots, a[i, n-i+1]$ 。

【输出格式】

输出文件 `briker.out` 中仅有一个正整数，表示被敲掉砖块的最大价值总和。

【输入输出样例】

输入：

```
4 5
2 2 3 4
8 2 7
2 3
4
```

输出：

```
19
```

解释：

敲掉第一行全部和第二行的 8。

数据范围：

- 对 10% 的数据， $n \leq 5$ ；
- 对 30% 的数据， $n \leq 10$ ；
- 对 50% 的数据， $n \leq 15$ ；
- 对 70% 的数据， $n \leq 30$ ；
- 对 100% 的数据， $n \leq 50$ ；其余数据范围在题面提及。

投资

-----ghy

题目描述:

现在商品社会发展得真快，连一向不懂经济的 Dragon 博士最近也决定下海了。他决定去做现下最流行的金融生意，于是他开了一家投资公司。但 Dragon 博士毕竟不懂经济呀，最近他已经连亏损了 3 个月了。但是正所谓吃一堑长一智，Dragon 博士发现：市场上美元和人民币之间的汇率总是上下波动，就是说：第 i 天时，一人民币可以兑换 $c[i]$ 美元，而一美元可以兑换 $d[i]$ 元人民币，但是美国人怕中国人的外汇储备太多，因此每次把美元兑换成人民币都要收取额外的 Cost 元人民币作为费用，当然把人民币在 兑换成美元是没人反对的，也就是说不须附加费用。于是 Dragon 博士就想呀，是否可以从中捞点油水呢？Dragon 的客户每天会向他提供 $w[i]$ 元人民币运转资金。每天剩余的资金（包括美元和人民币）都可以保存到第二天而无须附加费用。

输入文件:

第一行是两个整数 n , $cost$ ，分别指天数和额外费用。

后是 $n+1$ 行，第 $i+1$ 行是三个用空格格开的数： $w[i]$, $c[i]$, $d[i]$ 。

输出文件:

只有一行：第 n 天结束可以获得的人民币最大值（保留 4 位小数）。

数据范围:

$N \leq 2000$

$0 \leq cost \leq \text{maxlongint}; (\text{int})$

$0 \leq w[i] \leq 30$

$0 < c[i], d[i] \leq 10e5; (\text{real})$

$0 < c[i] * d[i] < 1$

所有输出均控制在 long int 之内。

输入样例:

```
3 1
10 0.1 9
10 0.01 20
10 0.1 9
```

输出:

```
39.0000
```

解释:

第一天， 全部人民币转美元；

第二天， 全部美元转人民币。

数据范围:

对 10% 的数据， $n \leq 5$

对 30% 的数据， $n \leq 20$

对 50% 的数据， $n \leq 100$

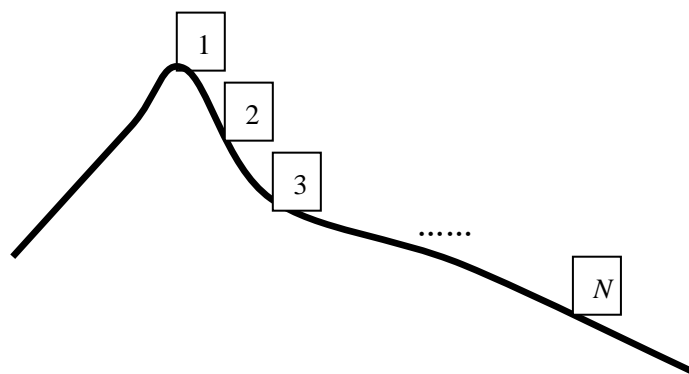
对 100% 的数据， $n \leq 2000$

其余数据范围在题面中有提及。

仓库建设

【问题描述】

L 公司有 N 个工厂，由高到底分布在一座山上。如图所示，工厂 1 在山顶，工厂 N 在山脚。



由于这座山处于高原内陆地区（干燥少雨），L 公司一般把产品直接堆放在露天，以节省费用。突然有一天，L 公司的总裁 L 先生接到气象部门的电话，被告知三天之后将有一场暴雨，于是 L 先生决定紧急在某些工厂建立一些仓库以免产品被淋坏。

由于地形的不同，在不同工厂建立仓库的费用可能是不同的。第 i 个工厂目前已有成品 P_i 件，在第 i 个工厂位置建立仓库的费用是 C_i 。对于没有建立仓库的工厂，其产品应被运往其他的仓库进行储藏，而由于 L 公司产品的对外销售处设置在山脚的工厂 N ，故产品只能往山下运（即只能运往编号更大的工厂的仓库），当然运送产品也是需要费用的，假设一件产品运送 1 个单位距离的费用是 1。假设建立的仓库容量都是足够大的，可以容下所有的产品。

你将得到以下数据：

- 工厂 i 距离工厂 1 的距离 X_i （其中 $X_1=0$ ）；
- 工厂 i 目前已有成品数量 P_i ；
- 在工厂 i 建立仓库的费用 C_i ；

请你帮助 L 公司寻找一个仓库建设的方案，使得总的费用（建造费用+运输费用）最小。

【输入文件】

输入文件 `storage.in` 第一行包含一个整数 N ，表示工厂的个数。接下来 N 行每行包含两个整数 X_i, P_i, C_i ，意义如题中所述。

【输出文件】

输出文件 `storage.out` 仅包含一个整数，为可以找到最优方案的费用。

【样例输入】

```
3
0 5 10
5 3 100
9 6 10
```

【样例输出】

```
32
```

【样例说明】

在工厂 1 和工厂 3 建立仓库，建立费用为 $10+10=20$ ，运输费用为 $(9-5)*3 = 12$ ，总费用 32。

如果仅在工厂 3 建立仓库，建立费用为 10，运输费用为 $(9-0)*5+(9-5)*3=57$ ，总费用 67，不如前者优。

【数据规模】

对于 20% 的数据， $N \leq 500$;

对于 40% 的数据， $N \leq 10000$;

对于 100% 的数据， $N \leq 1000000$ 。

所有的 X_i, P_i, C_i 均在 32 位带符号整数以内，保证中间计算结果不超过 64 位带符号整数。