NOIP2016模拟题 Day1

ZYF & WMJ

中文题目名称	小P的2048	小P的单调区间	小P的生成树
英文题目与子目录名	game	seq	mst
源程序名称	game.c/cpp/pas	seq.cpp/c/pas	mst.c/cpp/pas
输入文件名	game.in	seq.in	mst.in
输出文件名	game.out	seq.out	mst.out
每个测试点时限	1s	1s	2s
空间限制	256MB	256MB	256MB
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	无	有
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统

注意事项：

1、 考试时长为3.5个小时，请合理分配考试时间；

2、 文件名（源程序名和输入输出文件名）必须使用英文小写；

3、 请在选手目录下为每题单独建立子目录，并将源程序放在对应的子目录下；

4、 C/C++中函数main()的返回值必须是int，程序正常结束时的返回值必须是0；

5、 评测时不开启任何优化开关，题目时限以评测机配置为准。

小P的2048

【问题描述】

最近，小P迷上了一款叫做2048的游戏。这款游戏在一个$n \times n$的棋盘中进行，棋盘的每个格子中可能有一个形如$2^k (k \in \mathbf{N}^*)$的数，也可能是空的。游戏规则介绍如下：

1. 游戏开始时棋盘内将会生成两个数字，生成的数字仅可能为2或4；

2. 每次操作，玩家可以选择上、下、左、右四个方向进行平移；

3. 以向上平移为例，从上往下考虑每个不为空的格子，若上方与之相邻的格子为空，则将该格子上的数字移动至相邻格子。在一次**位移**中，每个数字会进行多次移动直到不能移动为止。

4. 以向上平移为例，从上往下考虑每个不为空的格子，若上方与之相邻的数字恰好与其相等，则这两个数字可以合并，新生成的数字为原来两个数之和。在一次**合并**中，每个数字只能与其它数合并一次，可以同时合并多对数字，但不能连续合并；

5. 每次操作由**位移+合并+位移**组成，若操作后棋盘局面发生变化，则该操作为有效操作，其**有效得分**为合并过程中所有新生成的数字之和；

6. 在每次操作之后，棋盘内都会新生成一个数字2或4，数字只会在空格子处生成；

7. 当棋盘被数字填满，玩家无法进行任何有效操作时，游戏结束，**游戏总得分**为所有操作的有效得分之和。

4		4	
		8	
	2	2	2
2	2	2	2

$\xRightarrow{\text{向左}}$

8			
8			
4	2		
4	4		

$\xRightarrow{\text{向下}}$

16	2		
8	4		

为了降低难度，小P对2048游戏进行了一些改动。在游戏开始前，小P会告诉你棋盘的初始状态，并且给你若干次操作。每次操作由方向变量、位置参数和一个数字组成，方向变量代表你在本次操作中的平移方向，给定的数字为本次操作过后将会新生成的数字的大小，而位置参数将决定生成数字的位置。若位置参数为K，**操作后**棋盘中空格子的数量为r，则新生成数字的位置为**从上到下**、**从左到右**第$(1 + K \bmod r)$个空格子。如果本次操作为无效操作，则游戏结束，而当所有操作都完成后，游戏同样结束。（**注意：改动后，游戏结束时棋盘不一定被数字填满。**）

4	①	2	②
③	2	8	2
4	④	⑤	⑥
8	⑦	2	2

$\xRightarrow{K=11}$

4		2	
	2	8	2
4		*	
8		2	2

现在小P问你，在游戏结束前你一共进行了多少次有效操作，最后你的游戏总得分是多少。

【输入格式】(game.in)

第一行为两个正整数n和m，分别表示棋盘的大小和操作的个数。

第二行为六个正整数x_1, y_1, v_1和x_2, y_2, v_2（$x_1, y_1, x_2, y_2 \leqslant n$，$v_1, v_2 \in \{2, 4\}$），分别代表游戏开始时，棋盘上两个数字的位置(行/列)和大小。行号从上往下编号，列号从左往右编号，编号均以1开始。

接下来m行，表示小P给你的m个操作。每行由三个自然数D_i, K_i, V_i组成，其中D_i代表本次操作的平移方向，0/1/2/3分别代表上/下/左/右。K_i为位置参数，V_i为操作后生成的数的大小。（$D_i \in \{0, 1, 2, 3\}$，$K_i < 2^{31}$，$V_i \in \{2, 4\}$）

【输出格式】(game.out)

输出共两行，每行一个正整数，分别代表你完成的有效操作数与游戏总得分。

【样例输入1】

```
3 6
2 1 2 3 2 4
1 14 2
1 6 4
2 10 2
0 3 2
2 17 2
3 14 4
```

【样例输出1】

4
12

【样例说明1】

2		
	4	

$\xrightarrow{\text{向下}}$

2		
2	4	

$\xrightarrow{\text{向下}}$

4	4	4

$\xrightarrow{\text{向左}}$

2		
8	4	

$\xrightarrow{\text{向上}}$

2	4	
8		
2		

四次有效操作后，棋盘无法继续向左平移，故游戏结束，总得分为$4+8=12$。

【样例输入输出2】

见附加样例文件中的**game/game0.in**和**game/game0.ans**。

【数据规模与约定】

对于10%的数据：$n=2$
对于40%的数据：$n\leqslant 4, m\leqslant 100$
对于100%的数据：$2\leqslant n\leqslant 8, 1\leqslant m\leqslant 100000$

小P的单调数列

【问题描述】

小P最近喜欢上了单调数列，因为他觉得单调的数列具有非常多优美的性质。经过小P复杂的数学推导，他计算出了一个单调增数列的艺术价值等于该数列中所有数的总和。并且以这个为基础，小P还可以求出任意一个数列的艺术价值，它等于将这个数列顺次划分为若干个**极长**单调区间（相邻两个单调区间的单调性必须不相同）后，每个单调区间中元素总和的平均值。比如对于数列 3 7 9 2 4 5，它将被划分为[3 7 9] [2] [4 5]，其艺术价值为$(19+2+9)/3=10$。由于小P特殊的审美观，他还要求划分出的第一个单调区间必须为单调增区间，也就是说，对于数列 10 9 8，它将被划分为[10] [9 8]，而不是[10 9 8]。

现在小P手里有一个长度为n的序列$\{a_i\}$，他想问你，这个序列的所有子序列中，艺术价值最大的是哪个子序列，输出其艺术价值。

注意：本题单调数列为严格单调，也就是说数列中的数必须严格上升或严格下降。

【输入格式】(seq.in)

输入的第一行为一个正整数n，代表序列的长度。

接下来的一行为n个正整数$a_1, a_2, \cdots, a_n$，代表序列中的n个数。

【输出格式】(seq.out)

输出仅有一个实数，表示子序列的最大艺术价值。

实数四舍五入，保留三位小数。

【样例输入1】

```
4
1 2 5 4
```

【样例输出1】

```
8.000
```

【样例输入2】

6
3 1 7 2 6 5

【样例输出2】

10.500

【样例说明】

对于第一个样例，最优的子序列为1 2 5，其价值为8。如果选择5 4的话，虽然和为9，但单调区间数为2，因为第一个单调区间必须为递增区间。

对于第二个样例，最优的子序列为3 7 6 5，其价值为$(3+7+6+5)/2=10.5$。

【数据规模与约定】

对于20%的数据：$n \leqslant 20$
对于40%的数据：$n \leqslant 200$
对于60%的数据：$n \leqslant 2000$
对于100%的数据：$2 \leqslant n \leqslant 100000$，$1 \leqslant a_i \leqslant 10^9$

小P的生成树

【问题描述】

小P是个勤于思考的好孩子，自从学习了最大生成树后，他就一直在想：能否将边权范围从实数推广到复数呢？可是马上小P就发现了问题，复数之间的大小关系并没有定义。于是对于任意两个复数z_1, z_2，小P定义$z_1 < z_2$当且仅当$|z_1| < |z_2|$。

现在，给出一张n个点m条边的简单无向带权图，小P想问你，如果按照他对复数大小关系的定义，这个图的最大生成树是什么？

【输入格式】(mst.in)

输入的第一行为两个正整数n和m，分别表示这个无向图的点数和边数。

接下来m行，每行四个整数$u, v, a, b(1 \leqslant u, v \leqslant n, -1000 \leqslant a, b \leqslant 1000)$，表示点$u$与点$v$之间有一条无向边，其边权为$a + bi$。

【输出格式】(mst.out)

输出仅有一个实数，它等于所求的最大生成树中所有边权之和的模长。

实数四舍五入，保留六位小数。

【样例输入1】

```
3 3
1 2 1 3
2 3 2 2
3 1 3 1
```

【样例输出1】

```
5.830952
```

【样例说明1】

显然，从该图三条边中任取两条便可以构成一棵生成树，这三棵生成树的边权之和分别为$z_1 = 3 + 5i$, $z_2 = 4 + 4i$, $z_3 = 5 + 3i$，其中$|z_2| = \sqrt{32} < |z_1| = |z_3| = \sqrt{34}$。

【样例输入2】

6 9
1 2 4 -1
2 3 4 1
3 4 -1 -5
1 5 -4 0
4 6 1 -6
2 6 -6 0
5 6 -7 5
2 4 7 1
1 4 -9 -5

【样例输出2】

27.459060

【样例输入输出3】

见附加样例文件中的**mst/mst0.in**和**mst/mst0.ans**。

【数据规模与约定】

对于10%的数据：$n \leqslant 6$
对于30%的数据：$n \leqslant 12$
对于另外20%的数据：每条边的边权均为实数
对于100%的数据：$n \leqslant 50, m \leqslant 200$，给定的无向图至少存在一个生成树

【提示】

简单无向图的定义为：没有任何重边和自环的无向图。
若复数$z_1 = a_1 + b_1 i$, $z_2 = a_2 + b_2 i$，则$z_1 + z_2 = (a_1 + a_2) + (b_1 + b_2)i$。
设复数$z = a + bi$，符号$|z| = \sqrt{a^2 + b^2}$表示该复数的模长。
若复数$z = a + bi$满足条件$b = 0$，则该复数为实数。