

模拟赛

demerzel

2019 年 3 月 27 日

题目名称	下棋	大树	数轴
题目类型	非传统题	传统题	传统题
可执行文件名	chess	tree	number
输入文件名	N/A	tree.in	number.in
输出文件名	N/A	tree.out	number.out
测试点数量	N/A	20	25
测试点时限	N/A	2.0s	4.0s
内存限制	N/A	256MB	256MB
编译选项	-O2 -std=c++11		

1 下棋

(white.cpp/black.cpp)

1.1 Description

这是一道非传统题。

五子棋是一个老少咸宜的棋类游戏。规则如下：在一个 $n \times m (n, m \leq 8)$ 的网格上，黑白双方轮流落子，黑棋先手。当某一方的五个棋子在横、竖、斜中某个方向上成一条相邻的线，则获胜，若棋盘下满了还没有哪一方获胜，则平局。

请实现一个会下棋的程序，接口及说明如下

```
namespace Black {  
    void Init(int n, int m);  
    int Turn(int pos);  
}  
  
namespace White {  
    void Init(int n, int m);  
    int Turn(int pos);  
}
```

其中 Init 函数会在最开始调用一次，用来告知你棋盘的大小是 n 行 m 列，游戏通过调用 Turn 函数来进行，该函数中的参数表示上一轮对方的落子点是在 $(\lfloor \frac{pos}{m} \rfloor + 1, pos \% m + 1)$ ，若 $pos = -1$ 则表示由你先手，而你的返回值则是表示接下来你的落子点，表示方法同上，注意你不能在已经有棋子的地方落子。

请实现上面的函数。具体来说，你需要提交两个文件 `black.cpp` 和 `white.cpp`，其中分别实现两个 namespace 中的函数。详情参见下发的 `black_sample.cpp` 和 `white_sample.cpp`。

注意：最好不要在 namespace 外实现代码，否则可能产生未知错误。

1.1.1 测试方法

通过在命令行中输入 `g++ chess.cpp black.cpp white.cpp -o chess -O2 -std=c++11` 来编译，之后通过输入 `./chess` 来进行测试。

1.1.2 评分方法

首先将让你的两个 ai 进行一场比赛，若总用时超过 10s 或着程序没有正常退出则计零分（内存不做特殊限制）

之后出题人将通过某种未知的方法为所有参赛选手的程序排好名次，之后名次为 x 的选手将获得 $100 * (0.96114447)^{x-1}$ 分。

2 大树

(tree.cpp)

2.1 Description

有一棵 n 个点的带边权的以 1 号点为根的有根树，小 Z 一开始站在根，有一个宝箱位于一号点之外的某一个点。

现在小 Z 可以在树上随便走，走过一条权值为 x 的边需要花费 x 秒，请告诉小 Z，在按照最优策略走的情况下，期望多少秒来到有宝箱的节点。输出这个值，误差 $1e-6$ 以内被认为是正确的。

2.2 Input format

第一行一个数 n 。

接下来 $n-1$ 行每行三个数 u, v, w ，表示 u 到 v 之间一条权值为 w 的边。

2.3 Output format

一行一个实数表示答案。

2.4 Sample input

4 1 2 1 1 3 1 3 4 2

2.5 Sample output

3.000

2.6 Explanation

最优的路线是 $1 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4$ ，期望花费为 3 秒。

2.7 Constraints

对于测试点 1 ~ 5: $n \leq 8$ 。

对于测试点 6 ~ 7: 所有点的父亲都是 1 号点。

对于测试点 8 ~ 11: 每个点最多两个儿子。

对于测试点 12 ~ 20: $n \leq 100000$ 。

对于所有测试点，边权 ≤ 1000 。

3 数轴

(number.cpp)

3.1 Description

数轴上有 n 个区间，第 i 个区间是 $[L_i, R_i]$ ，你可以进行不超过 k 次操作，每次操作可以选定一个区间，将其左移或右移一个单位长度。

操作完之后，求出所有区间的交，最终收益为交中的“好数”的个数。求最大收益。

若一个正数的十进制表示中只有“4”和“7”出现，那么这就是一个好数。比如 474，4477 是好数，475，233 则不是。

3.2 Input format

第一行两个数 n, k 。

接下来 n 行每行两个数 L_i, R_i 。

3.3 Output format

一行一个数表示答案。

3.4 Sample input

```
4 7
1 4
6 9
4 7
3 5
```

3.5 Sample output

```
1
```

3.6 Constraints

对于测试点 1 ~ 4: $n, k \leq 8$

对于测试点 5 ~ 12: $n \leq 10^3$, $R_i \leq 10^9$ 。

对于测试点 13 ~ 16: $n \leq 10^5$ ，存在一个区间满足 $L_i = R_i$ 。

对于测试点 17 ~ 25: $n \leq 10^5$, $k \leq 10^{18}$, $R_i \leq 10^{18}$ 。

对于所有测试点， $L_i \leq R_i$ 。