

20170120 模拟赛题解

共 3 道题目，时间 3.5 小时

题目名称	恐狼后卫	交错和查询	直线斯坦纳树
文件名	wolf	sum	rsmt
输入文件	wolf.in	sum.in	rsmt0-9.in
输出文件	wolf.out	sum.out	rsmt0-9.out
时间限制	1000ms	1000ms	N/A
内存限制	256mb	256mb	N/A
测试点数目	10	10	10
测试点分值	10	10	10
是否有部分分	否	否	有
题目类型	传统	传统	提交答案

评测环境

操作系统: Windows 8.1

CPU: Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz

系统内存: 8.00GB

评测工具: cena 0.8.1

Problem 1 恐狼后卫(wolf.cpp/c/pas)

【题目描述】

著名卡牌游戏《石炉传说》中有一张随从牌：恐狼后卫。恐狼后卫的能力是使得相邻随从的攻击力提高。

现在有 n 张恐狼后卫顺序排成一排，第 i 只恐狼后卫的攻击力为 $a[i]$ ，血量为 $h[i]$ ，提升相邻随从的攻击力值为 $b[i]$ 。

你的攻击力为 atk ，每次攻击你可以选择一只存活的恐狼后卫，减少其血量值 atk 。若其血量小于等于 0，则该恐狼后卫死亡。当某只恐狼后卫死亡时，其左右两侧（若存在）的恐狼后卫会靠拢并成为相邻关系。

在攻击第 i 只恐狼后卫时，除了要承受这只恐狼后卫自身的攻击力 $a[i]$ 之外，还要承受与其相邻的 2 张恐狼后卫的提高攻击力值 $b[i-1]$ 和 $b[i+1]$ （若存在）。

你的任务是承受最少的总伤害杀死所有恐狼后卫，输出需承受的伤害值。

【输入格式】

第一行一个正整数 n ，表示恐狼后卫的数量。

第二行一个正整数 atk ，表示你的攻击力。

以下 n 行，每行 3 个值： $a[i]$ 、 $b[i]$ 、 $h[i]$ ，分别表示第 i 只恐狼后卫自身的攻击力值、提升相邻随从的攻击力值、血量值。

【输出格式】

一个整数，表示杀死所有恐狼后卫需要承受的最少伤害值。

【样例输入】

```
3
1
8 1 6
3 5 7
4 9 2
```

【样例输出】

```
94
```

【数据范围】

对于 30% 的数据， $n \leq 10$

对于另外 30% 的数据， $n \leq 100$ ， $h[i] = 1$

对于 100%的数据, $n \leq 400$, atk 、 $a[i]$ 、 $b[i]$ 、 $h[i] \leq 1000$

【解题报告】

对于 30%的数据, $n \leq 10$, 直接穷举所有情况即可。

对于另外 30%的数据, $n \leq 100$, $h[i] = 1$

首先本题中的 $a[]$ 直接求和即可, 只需要考虑 $b[]$ 的影响。常规的解决该类问题的思路是区间 DP, 枚举第一个去掉的位置并将问题转化为左右两个子区间的问题, 而本题的关键是在杀死一只恐狼后卫后, 其相邻两侧的恐狼后卫会靠拢, 且恐狼后卫能提升相邻的随从的攻击力。所以如若枚举区间内第一个杀死的恐狼后卫, 左右两区间的相邻的恐狼会互相影响, 所以并不是一个子问题。此时应反其道而行之, 枚举最后一只杀死的狼, 在这种情况下左右两个子区间被这最后一只狼隔开, 就成为了 2 个子问题。

详细描述一遍, 设 $f[i][j]$ 为消灭第 i 只到第 j 只恐狼的最优代价(被第 $i-1$ 只狼和第 $j+1$ 只狼包围), 枚举 k 作为最后一只被杀死的狼, 则在杀死这第 k 只狼时, 将会受到第 $i-1$ 只狼和第 $j+1$ 只狼的攻击力提升。

$$f[i][j] = \min(f[i][k-1] + f[k+1][j] + b[i-1] + b[j+1])$$

这是 $O(n^3)$ 的 DP, 可以顺利解决。

对于 100%的数据, 恐狼有血量值, 可以先根据玩家的攻击力 atk 换算出攻击次数。用贪心思想易得, 存在一个最优方案: 攻击某只恐狼必连续攻击至死, 再攻击别的恐狼。所以 DP 方程略微修改为:

$$f[i][j] = \min(f[i][k-1] + f[k+1][j] + (b[i-1] + b[j+1]) * times[k])$$

其中 $times[k]$ 为杀死第 k 只狼需要的攻击次数。

Problem 2 交错和查询 (sum.cpp/c/pas)

【题目描述】

无限循环数字串 S 由长度为 n 的循环节 s 构成。设 s 为 12345 ($n=5$)，则数字串 S 为 123451234512345...

设 S_i 为 S 的第 i 位数字，在上面的例子中， $S_1=1$ ， $S_2=2$ ， $S_6=1$ 。

设 S 的一个子串 $S[l, r]$ 的交错和为 $\text{sum}(l, r)$ ：

$$\text{sum}(l, r) = S_l - S_{l+1} + S_{l+2} - S_{l+3} + \cdots + (-1)^{r-l} S_r$$

如 $\text{sum}(2, 7) = 2 - 3 + 4 - 5 + 1 - 2 = -3$

现给定循环节 s ，要求支持两种操作：

1 pos digit:

修改循环节 s 上的某一位，即将 s_{pos} 改为 digit。

2 l r:

求 $S[l, r]$ 内所有子串的交错和的和，即

$$\text{ans} = \sum_{l \leq i \leq j \leq r} \text{sum}(i, j)$$

输出 ans 对 10^9+7 的模。

【输入格式】

第一行一个整数 n ，表示循环节 s 的长度。

第二行一个长度为 n 的数字串，表示循环节 s 。

第三行一个整数 m ，表示操作次数。

以下 m 行，每行 3 个整数。

若第一个数为 1，表示是修改操作 1 pos digit。

若第一个数为 2，表示是询问操作 2 l r。

【输出格式】

对于每个询问操作输出一行，表示答案。

【样例输入】

```
5
12345
5
```

2 1 5
2 6 10
1 3 5
2 1 5
2 1 6

【样例输出】

19
19
25
36

【数据范围】

对于 10%的数据点, $n, m \leq 50$;
对于 20%的数据点, $n, m \leq 1000$;
对于 40%的数据点, $1 \leq l \leq r \leq n$;
对于 100%的数据点, $n, m \leq 200000$; $1 \leq l \leq r \leq 10^{18}$; $1 \leq pos \leq n$;
 $0 \leq digit \leq 9$;

【解题报告】

考察要点: 线段树、分段思想、循环节
涉及要点: 快速 I/O
题目原型: 2014 ACM/ICPC 牡丹江赛区网络赛

1) 对于 10%的数据点, $n, m \leq 50$;

按照题面描述模拟即可。
时间复杂度: $O(n^3m)$

2) 对于 20%的数据点, $n, m \leq 1000$;

对于询问操作 $[l, r]$, 考察 S_i 对 ans 的贡献。
通过计算每个跨过 S_i 的 $\text{sum}(x, y)$ (即 $x \leq i \leq y$) 中 S_i 的正负性, 可以总结出如下规律:
a) 若 i 与 l 同奇偶, 则 S_i 对 ans 的贡献是 $(r-i+1) * S_i$;
b) 若 i 与 l 不同奇偶, 则 S_i 对 ans 的贡献是 0。

如询问区间内的数是 67312, 则 $ans = 6 * 5 + 7 * 0 + 3 * 3 + 1 * 0 + 2 * 1$;

又如询问区间内的数是 905173, 则 $ans = 9 * 6 + 0 * 0 + 5 * 4 + 1 * 0 + 7 * 2 + 3 * 0$ 。

简而言之，ans 即为数字串 S 各位与一等差数列的乘积，同时统计时还要依据下标区分奇偶。

依据此性质，直接用数组维护数字串， $O(1)$ 维护修改操作， $O(n)$ 解决一个询问操作。

时间复杂度： $O(nm)$

- 3) 对于 40% 的数据点， $1 \leq l \leq r \leq n$;

此时 n, m 的大小上升为 200000，但询问区间 $[l, r]$ 只会落在第一个循环节内（即可认为是没有循环节的有限数字串）。使用线段树维护 4 种信息，分别是区间内奇/偶位置的和、区间内奇/偶位置与一等差数列的乘积的和。修改和查询操作的复杂度均为 $O(\log n)$ 。

时间复杂度： $O(m \log n)$

- 4) 对于 100% 的数据点， $n, m \leq 200000$ ； $1 \leq l \leq r \leq 10^{18}$;

此时询问区间 $[l, r]$ 可能横跨多个循环节。借助类似于分段算法的思想，将整个询问区间拆成三份：前缀、中间和后缀。其中前缀是询问区间 $[l, r]$ 中第一小段不完整的部分（可为空）；中间部分是询问区间 $[l, r]$ 中横跨的所有完整的循环节；后缀是询问区间 $[l, r]$ 中最后一小段不完整的部分（可为空）。前缀和后缀部分的答案按照 60% 的数据点中的做法查询线段树即可。而中间部分的答案需要计算出横跨过的完整的循环节个数，并根据 n 与循环节个数的奇偶性的不同辅以不同的计算公式，其推导比较简单但有不少细节，可参见代码。

此外，本题可以通过将循环节延长一倍使得 n 必定为偶数，这样做就可以避免 n 为奇数时还需使用不同的计算公式。不过这么做的话，线段树的规模扩大一倍，而且一个修改操作需要修改 2 次线段树。算法常数更大，选手可以通过快速 I/O 等手段改善程序性能。

时间复杂度： $O(m \log n)$

预期表现：

直接模拟：10 分

根据性质暴力：20 分

线段树：40 分

线段树+循环节（分段思想）：100 分

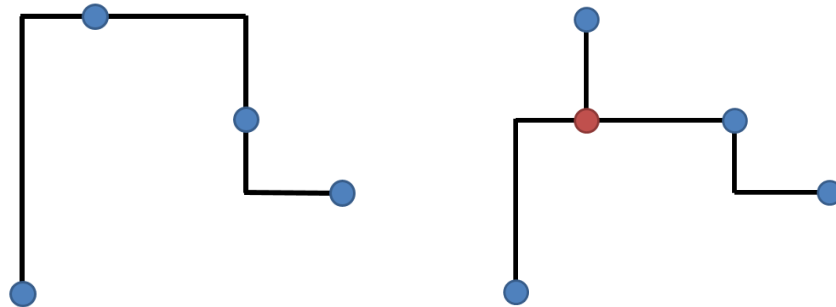
Problem 3 直线斯坦纳树(rsmt0-9.out)

【题目描述】

给定二维平面上 n 个整点，求该图的一个直线斯坦纳树，使得树的边长度总和尽量小。

直线斯坦纳树：使所有给定的点连通的树，所有边必须平行于坐标轴，允许在给定点外增加额外的中间节点。

如下图所示为两种直线斯坦纳树的生成方案，蓝色点为给定的点，红色点为中间节点。



【输入格式】

第一行一个整数 n ，表示点数。

以下 n 行，每行两个整数表示点的 x, y 坐标。

【输出格式】

第一行一个整数 m ，表示中间节点的个数。

必须满足 $m \leq 10 * n$

以下 m 行，每行 2 个整数表示中间节点的坐标。

以下 $n+m-1$ 行，每行 2 个整数 u, v 表示编号为 u 和 v 的两个节点之间有边直接相连，边的权值为两点的曼哈顿距离，即 $|x[u] - x[v]| + |y[u] - y[v]|$

输入数据给定的点的编号为 $1..n$ ，选手输出的中间节点的编号为 $n+1..n+m$

【样例输入】

```
4
-1 0
0 -1
1 0
0 1
```

【样例输出】

```
1
0 0
1 5
2 5
3 5
4 5
```

【格式说明】

选手可使用 test.exe 对数据进行测试

在命令行下使用 “test x” 即可测试 rsmtx.in 和 rsmtx.out

（注：x 必须是一个字符，即 “test 1234” 等价于 “test 1”）

使用 “test @” 即可测试样例

测试结束前 1 小时发布 gen.exe，选手将该程序与所有输入输出文件（rsmt0..9.in / .out）放在同一目录，运行后生成 rsmt-upload.cpp，测试结束时上交 rsmt-upload.cpp 而非 rsmt0..9.out

若选手没有完成某个测试点，可不必生成该测试点的输出文件，gen.exe 会自动跳过该测试点。

【得分说明】

若选手的解合法，则该数据点至少得 1 分

设选手的答案为 ans，标程的答案为 std。

若 $ans \geq std$ ，则选手得分为 $10 * (std/ans)^w$ ，其中 w 是该测试点的得分系数。

若 $ans < std$ ，则所有选手中的最优解作为 std，该选手得 12 分，其他选手的得分为 $12 * (std/ans)^w$

【解题报告】

前四个数据点，有 $n \leq 8$ ，使用人工构造、随机化搜索算法、状态压缩动态规划等均可求出最优解。

第 5 个至第 8 个数据点，所有的点排布在两行上，即点的 y 坐标只有 2 种取值。故我们可以根据 x 坐标的顺序从左至右进行状态压缩 DP，状态为当前列上的两个点的连通情况和是否有需要加入斯坦纳树的点，共四种情况：两个点属于一个连通块；两个点不属于同一个连通块，此时有 3 种情况：两个块内都存在需要加入斯坦纳树的点，其中某个块存在需要加入斯坦纳树的点（因为有 2 个连通块，故这种情况有 2 种）。依据这种状态设计进行状压 DP 即可，值得注意的是本题需要输出方案，故还需保存 DP 过程中的转移路径，并编写程序输出。

最后 2 个数据点为纯随机数据点，适合模拟退火等随机化搜索算法，或一些

贪心+构造的策略，是开放性的问题，也可以参考《FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design》等文献提供的方法。