

Contest 9 By Group S

Problems

- A. String Roughness(25/78)
- B. A Two Floors Dungeon(5/25)
- C. Area Folding(0/3)
- D. Connect(0/0)
- E. Rating System(21/103)
- F. House Moving(13/45)

Problem	AC	WA	PE	RTE	FPE	SF	NZEC	TLE	MLE	OLE	CE	Submit
A	25	48	0	0	0	0	0	3	0	0	2	78
B	5	7	0	8	0	2	0	0	3	0	0	25
C	0	3	0	0	0	0	0	0	0	0	0	3
D	0	0	0	0	0	0	0	0	0	0	0	0
E	21	81	0	1	0	0	0	0	0	0	0	103
F	13	22	0	0	0	2	0	6	1	0	1	45
Summary	64	161	0	9	0	4	0	9	4	0	3	254

Problem A: String Roughness

问题概要

- 定义Roughness为这个字符串中个数最多的字符和个数最少（不能为0）的字符数量之差。可以删除其中 $0 \sim n$ 个字符，求删除后最小的Roughness

解法：

批量删除数量最少的字符

或单独删除数量多的某个字符

- 数据范围小（ $|s| \leq 50$ ），暴力

Problem B:
A Two Floors Dungeon

问题概要

- 有2层的迷宫
- 拨动开关会使某些格子上下移动
 - 当然自己所在的格子也有可能移动
- 求起点到终点的最短时间

解法

- {行·列·层} × {ON/OFF} 为状态BFS
- 每个格子现在在哪一层可以通过对如下状态进行位压缩
 - 每个开关是否可以移动这个格子
 - 每个开关的 ON /OFF状态计算出来
- 要注意开关可能会移动当前所在的格子

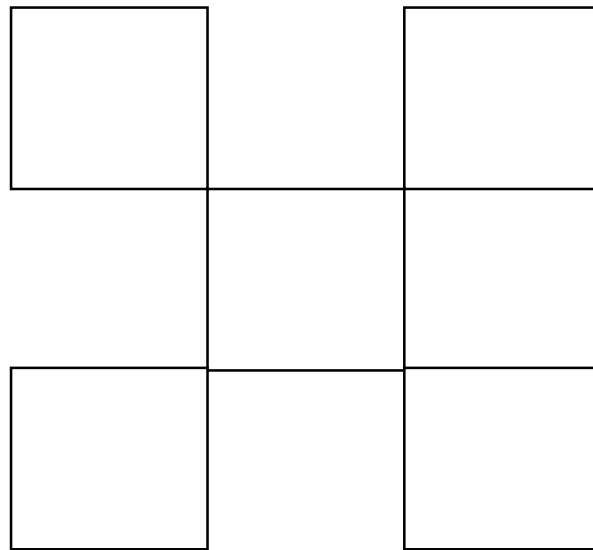
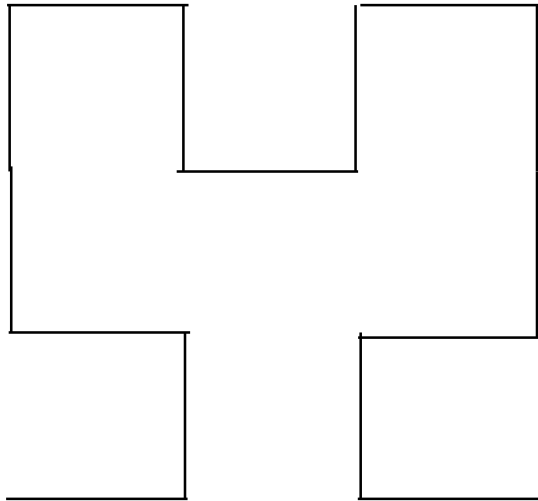
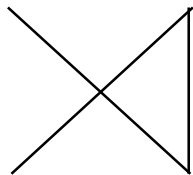
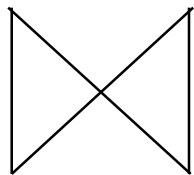
Problem C: Area Folding

问题

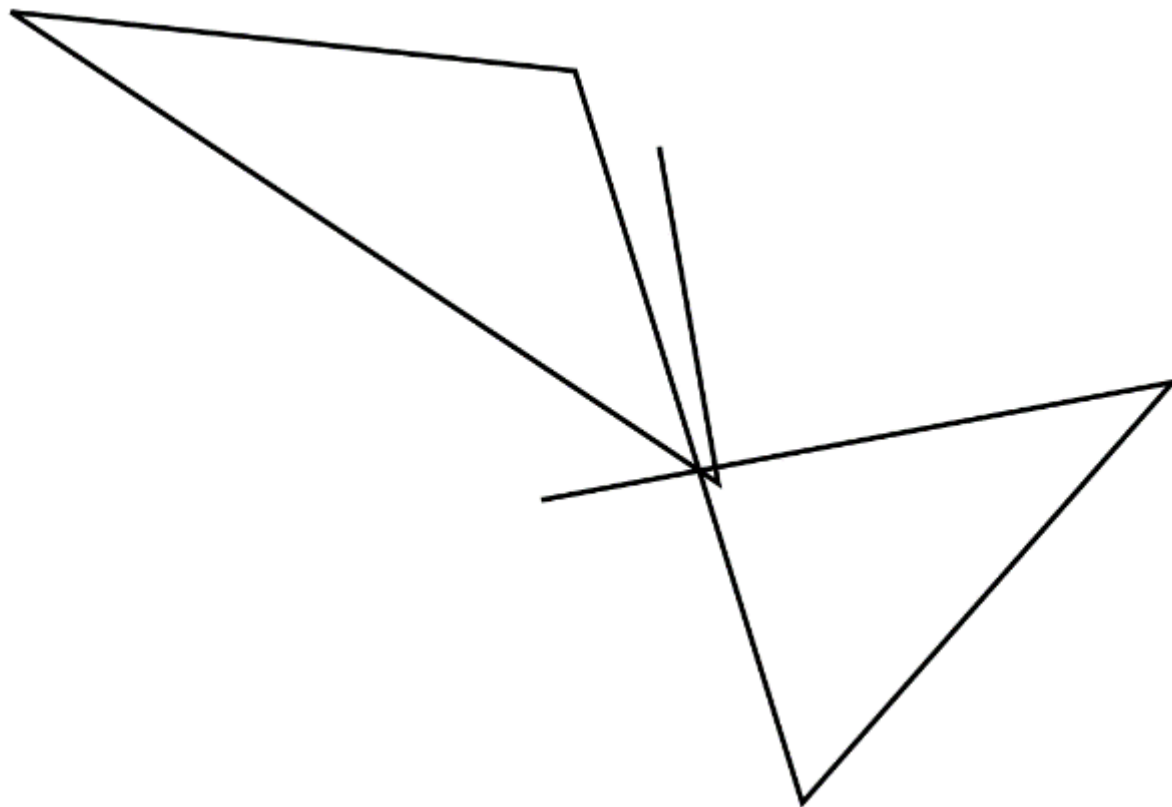
- 有一条 n 个顶点的折线
- 求折线形成的封闭区域的面积

- $n \leq 100$
- $|\text{坐标}| \leq 100,000$
- 构成折线的每条线段除了端点都和其他的顶点距离0.01以上
 - 这是为了防止因为产生误差而导致错误地判断了是否是封闭区域

样例



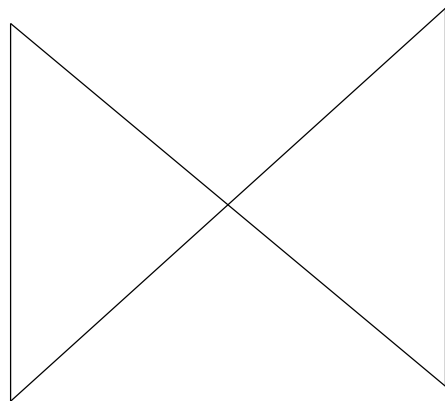
样例



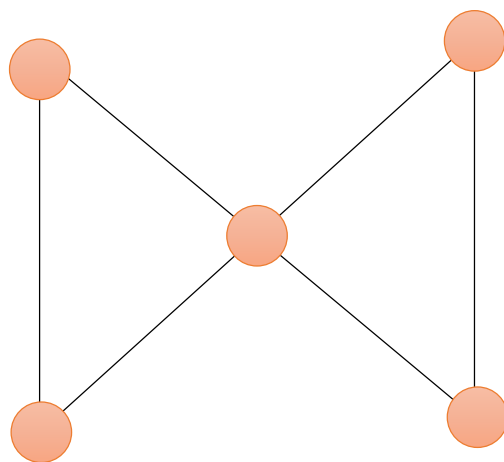
解法

- 把图形分解为多个多边形来计算
- 适当排列线段然后构图
- 对于每个关键点把边按照倾斜角排序
- 每次绕着逆时针方向走构建出所有的多边形
- 多边形的面积为正则加到答案里，是负则是外部区域因此丢弃
- 实现起来比较麻烦
 - 计算过程会有比较大的数产生需要注意误差

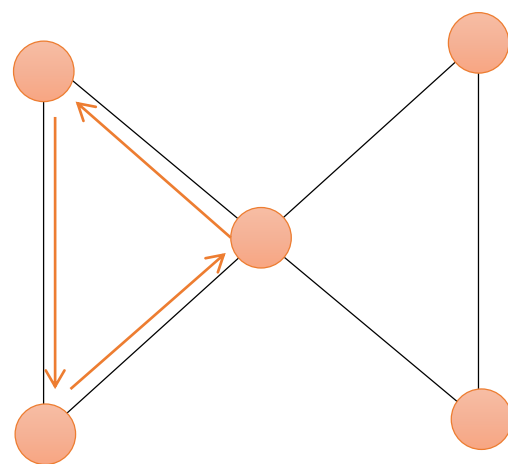
把折线



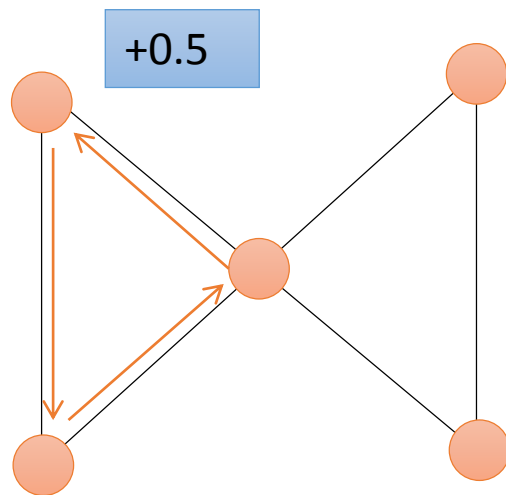
提取关键点构图



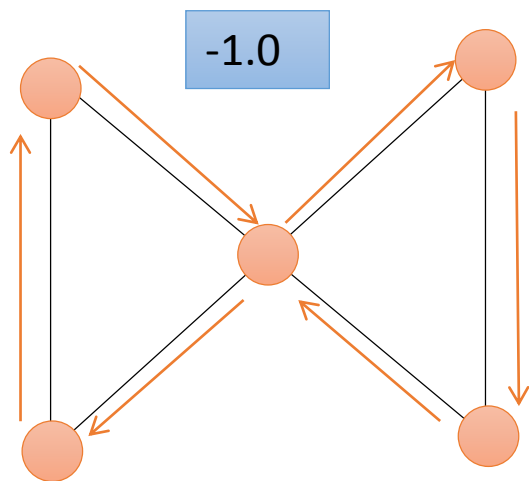
按照逆时针绕



如果面积为正则加到答案里



如果是负的就丢弃



Problem D: Connect

Connect

- 给出 R 个字符串，要求摆放在一个 $R * C$ 的网格图里，每行放一个字符串，字符串在摆放时可以随意添加空格在前后或是字符之间，但不能调换原先的字符顺序。
- 对于网格图上相邻的两个格子，如果字符相同，则 $ans + 1$
- 求最大的 ans 可能为多少

问题概要

填字游戏。R*C的方格每行填一个指定单词，字母间可以有空格。

A	C		M
I	C	P	C

最后得分：每个格子的每个相邻格字母相同总分+1，相邻的定义是有公共边。

数据范围 $R \leq 128$ $C \leq 16$

- $dp[i][j][mask]$ 表示确定了前 i 行，并且在第 i 行确定了前 j 列，此时最后 C 个被确定的格子（可能跨行）在 $mask$ 的位置上有字符时的 dp 值

P	R	O	B	L	E	M	F	
C		O	N	N	E	C		T

- 类
- 上面的例子里暖色的格子表示已经处理完的格子，目前要处理的决策是 $dp[1][3][(111101011)_2] \rightarrow dp[1][4][(111010111)_2]$

- 通过 j 可以确定当前这行用了哪些字符，即将摆放什么字符
- 另外还可以通过 j 知道 `mask` 里包含了上面那行的哪些字符
- 从而知道转移时是 $+0$ 还是 $+2$ (和左边相同或者和上面相同) 还是 $+4$ (两边都相同)
- 注意到关系是双向的所以不是 $+1 +2$ 而是 $+2 +4$

E.Rating System

- 给出 N 个人的 rating R_i , 挥发度 V_i , 比赛排名 P_i 和账户余额 B_i
- 求比赛结束时的各人 rating
- 注意到由于系统 bug, E_i 会被设置为 P_i 的值
- 所以那些公式都是没有用的!
- 实际上 $\text{New}R_i = R_i + B_i$
- 解法: 用 `pair<int, string>` 排序, 然后输出

Problem F: House Moving

问题概要

- 数组A有n个互不相同的元素，问移动某些元素使它变成递增序列移动的最小花费。
- 移动第i个元素的费用是A[i]

解法

- 类似于求最大递增序列，目标由序列长度最长变为序列中元素和最大。
- $V[i] = \max(V[k] + A[i] \mid A[k] < A[i])$
- $O(n^2)$ 会超时
- 保存有序的数组 $(A[k_1], V[k_1]), (A[k_2], V[k_2]), \dots$
 $A[k_i] < A[k_j] \& \& V[k_i] < V[k_j] \forall i < j$
- 转移操作复杂度为 $O(\log(n))$
- 维护数组总复杂度 $O(n \log(n))$