

NOI2007

Day 1

解题报告



CmYkRgB123

<http://www.cmykrgb123.com>

河南省实验中学 郭家宝

2008.4.3

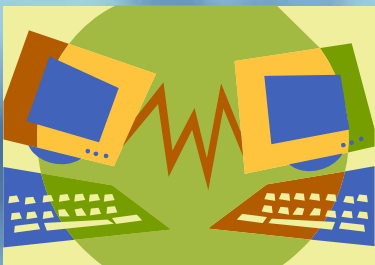
CmYkRgB123

声明

- 水平有限
- 时间仓促
- 资料不足
- 请多指教

题目一览

- 社交网络
- 货币兑换
- 调兵遣将



社交网络

CmYkRgB123

【问题描述】

- 在一个社交圈子里有 n 个人，人与人之间有不同程度的关系。我们将这个关系网络对应到一个 n 个结点的无向图上，两个不同的人若互相认识，则在他们对应的结点之间连接一条无向边，并附上一个正数权值 c ， c 越小，表示两个人之间的关系越密切。

【问题描述】

- 我们可以用对应结点之间的最短路长度来衡量两个人 s 和 t 之间的关系密切程度，注意到最短路径上的其他每个结点 v 都为 s 和 t 的联系提供了便利，即所有结点 v 对于 s 和 t 之间的联系有一定的重要程度。
- 我们可以通过统计经过一个结点 v 的最短路径的数目来衡量该结点在社交网络中的重要程度。

【问题描述】

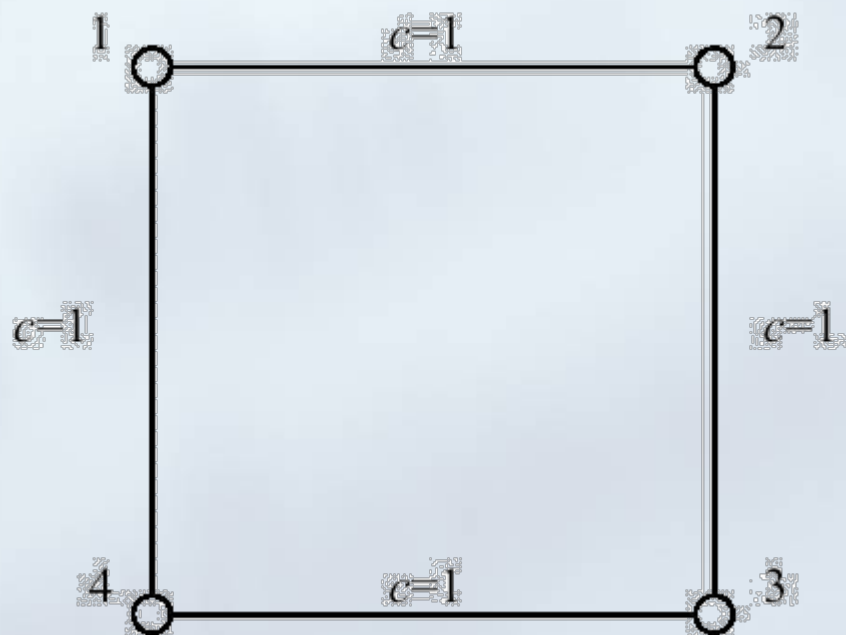
- 考虑到两个结点A和B之间可能会有多条最短路径。我们定义v的重要程度 $I(v)$ 的定义如下：
令 $C_{s,t}$ 表示从s到t的不同的最短路的数目，
 $C_{s,t}(v)$ 表示经过v从s到t的最短路的数目，即

$$I(v) = \sum_{s \neq v, t \neq v} \frac{C_{s,t}(v)}{C_{s,t}}$$

【问题描述】

- 为结点 v 在社交网络中的重要程度。
为了使 $I(v)$ 和 $C_{s,t}(v)$ 有意义，我们规定需要处理的社交网络都是连通的无向图，即任意两个结点之间都有一条有限长度的最短路径。
- 现在给出这样一幅描述社交网络 s 的加权无向图，请你求出每一个结点的重要程度。

举例说明



- 对于1号结点而言，只有2号到4号结点和4号到2号结点的最短路经过1号结点。
- 而(2, 4) (4, 2) 之间的最短路径又都只有2条。且都只有1条经过1号结点的重要程度为 $1/2+1/2=1$ 。

输入输出

- 输入文件中第一行有两个整数， n 和 m ，表示社交网络中结点和无向边的数目。在无向图中，我们将所有结点从1到 n 进行编号。
接下来 m 行，每行用三个整数 a ， b ， c 描述一条连接结点 a 和 b ，权值为 c 的无向边。注意任意两个结点之间最多有一条无向边相连，无向图中也不会出现自环（即不存在一条无向边的两个端点是相同的结点）。
- 输出文件包括 n 行，每行一个实数，精确到小数点后3位。第 i 行的实数表示结点 i 在社交网络中的重要程度。

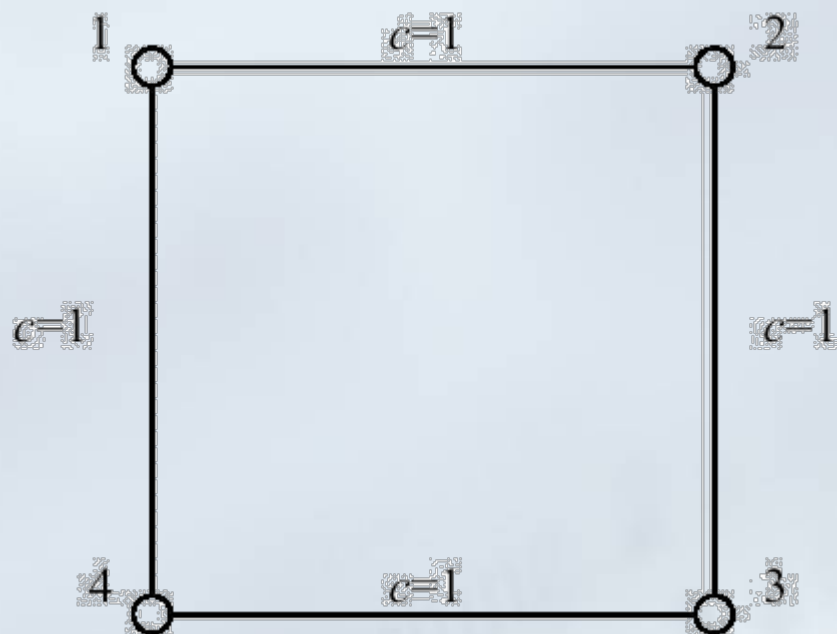
样 例

- 【样例输入】

- 4 4
1 2 1
2 3 1
3 4 1
4 1 1

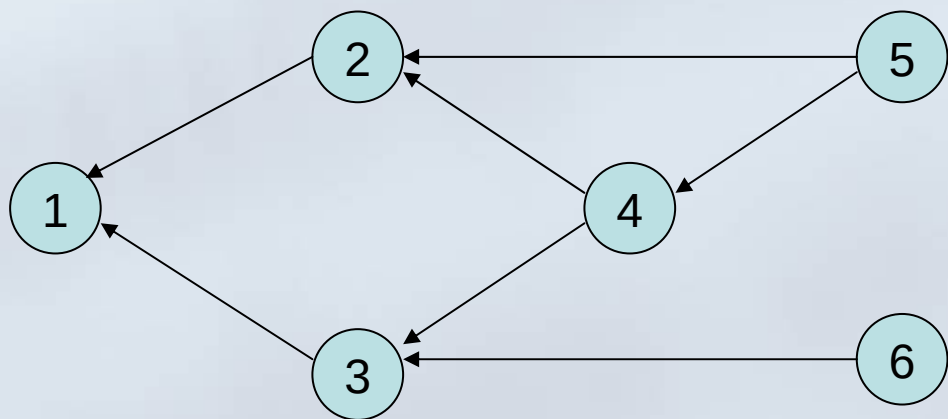
- 【样例输出】

- 1.000
1.000
1.000
1.000



算法分析1

- 看完这道题我首先的想法就是，求出所有点间最短路径，然后再计算统计。
- 方法就是在计算最短路的时候将每条路径保存下来，使用有向树形图存储。



1	2	3	4	5	6
{0}	{1}	{1}	{2, 3}	{2, 4}	{3}

6,3,1 4,2,1
5,2,1 4,3,1
5,4,2,1 3,1
5,4,3,1 2,1

算法实现

1. 初始化，构图
2. 以 i 为源计算 $I(v)$ $i=1..n$
 1. 计算以 i 为源的最短路(SPFA)
 2. 遍历交叉链表，统计每个点经过的次数和路径条数
 3. 累和 $I(v)$ $v=1..n$
3. 输出 $I(v)$ 的值

SPFA实现

```
void spfa(long k, long *sp)
```

```
1.      将源点插入队列
2.      while (队列不为空)
3.      {          i=队首元素出队
4.          for (枚举i能连接到的每一个点j)
5.              if (sp[i]+v<sp[j]) //判断能否松弛 改变父亲
6.              {          sp[j]=sp[i]+dis[i][j]; //松弛
7.                          father[j][ father[j][0]=1 ]=i;
8.                          如果j不在队列中则把j加入队列
9.              }
10.         else if (sp[i]+dis[i][j]==sp[j]) //判断添加平行父亲
11.         {          father[j][ ++father[j][0] ]=i;
12.                     如果j不在队列中则把j加入队列
13.         }
14.     }
```

统计实现

void count()

- for (遍历每一个顶点i, 计算从每个顶点开始的最短路径)
- { spfa(i);
- for (遍历每一个顶点j, 统计从i到j的所有最短路径)
- { get(i, j);
- //遍历树形图, 把每个顶点出现的次数累计到ki[k]上
- for (遍历每个顶点, 把统计到的数量加到I(v)上)
- I[k]+=ki[k] / cntd;
- //每个顶点出现次数除以(i, j)所有最短路径条数
- }
- }

算法结论

- 时间复杂度 $O(n^3)$
- 编程复杂度 较高
- 代码长度 193 行
- 最后得分 60
- 烂算法!!!

程序在这里



Network2.cpp

算法分析2

- 更好的算法 Floyd + DP
- 记 s 至 t 的最短路径长度为 $D_{s;t}$, s 至 t 的最短路径条数为 $F_{s;t}$ 。易知:
- $C_{s;t}(v) = F_{s;v} \times F_{v;t}; (D_{s;v} + D_{v;t} = D_{s;t})$
- $C_{s;t}(v) = 0; (D_{s;v} + D_{v;t} > D_{s;t})$
- 故只需做一遍Floyd, 并在做的过程中记录下 $F_{s;t}$ 即可。

算法要点

- 要在Floyed的过程中计算好最短路的数目，而且要分为完全松弛和平行最短路两种情况。

```
void floyed()
```

```
1. {   long i, j, k;  
2.     for (k=1; k<=n; k++)  
3.       for (i=1; i<=n; i++)  
4.         for (j=1; j<=n; j++)  
5.           if (dis[i][k]+dis[k][j]<dis[i][j])  
6.             {   dis[i][j]=dis[i][k]+dis[k][j];  
7.                 path[i][j]=path[i][k]*path[k][j];  
8.             }  
9.           else if (dis[i][k]+dis[k][j]==dis[i][j])  
10.             path[i][j]+=path[i][k]*path[k][j];  
11. }
```

算法要点

- 计算 $I(v)$ 时要注意必须是最短路，而且不包括两个端点

```
void count()
```

```
1. {  
2.   for (k=1;k<=n;k++)  
3.     for (i=1;i<=n;i++)  
4.       for (j=1;j<=n;j++)  
5.         if (dis[i][k]+dis[k][j]==dis[i][j]&&(i!=j)&&(i!=k)&&(j!=k))  
6.           I[k]+=path[i][k]*path[k][j]/(double)path[i][j];  
7.           //如果k是i, j最短路径上一点，那么计算 $I(v)$   
8. }
```

算法结论

- 时间复杂度 $O(n^3)$
- 编程复杂度 低
- 代码长度 87 行
- 最后得分 100
- 正确算法!!!

程序在这里



Network.cpp

总结

- 本题是这次NOI2007中最简单的一道题目，可以把它理解为送分题吧。
- 考察的知识点有：最短路算法，简单动态规划。
- 尽管被称为送分题，在看别人的解题报告之前我还是没有想出算法2，自己写了算法1。



货币兑换

CmYkRgB123

【问题描述】

- 小Y最近在一家金券交易所工作。该金券交易所只发行交易两种金券：A纪念券（以下简称A券）和B纪念券（以下简称B券）。每个持有金券的顾客都有一个自己的帐户。金券的数目可以是一个实数。每天随着市场的起伏波动，两种金券都有自己当时的价值，即每一单位金券当天可以兑换的人民币数目。我们记录第K天中A券和B券的价值分别为 A_K 和 B_K （元/单位金券）。

【问题描述】

- 为了方便顾客，金券交易所提供了一种非常方便的交易方式：比例交易法。
比例交易法分为两个方面：
 - a) 卖出金券：顾客提供一个 $[0, 100]$ 内的实数 OP 作为卖出比例，其意义为：将 $OP\%$ 的A券和 $OP\%$ 的B券以当时的价值兑换为人民币；
 - b) 买入金券：顾客支付 IP 元人民币，交易所将会兑换给用户总价值为 IP 的金券，并且，满足提供给顾客的A券和B券的比例在第 K 天恰好为 $Rate_K$ ；

【问题描述】

- 例如，假定接下来3天内的 A_k 、 B_k 、 $Rate_k$ 的变化分别为：

时间	A_k	B_k	$Rate_k$
第一天	1	1	1
第二天	1	2	2
第三天	2	2	3

【问题描述】

- 假定在第一天时，用户手中有100元人民币但是没有任何金券。用户可以执行以下的操作：

时间	操作	人民币	A券数量	B券数量
开户	无	100	0	0
第一天	买入100元	0	50	50
第二天	卖出50%	75	25	25
第二天	买入60元	15	55	40
第三天	卖出100%	205	0	0

【问题描述】

- 注意到，同一天内可以进行多次操作。
小Y是一个很有经济头脑的员工，通过较长时间的运作和行情测算，他已经知道了未来N天内的A券和B券的价值以及Rate。他还希望能够计算出来，如果开始时拥有S元钱，那么N天后最多能够获得多少元钱。

【问题描述】

- 【输入格式】
- 第一行两个正整数 N 、 S ，分别表示小 Y 能预知的天数以及初始时拥有的钱数。接下来 N 行，第 K 行三个实数 A_K 、 B_K 、 $Rate_K$ ，意义如题目中所述。
- 【输出格式】
- 只有一个实数 $MaxProfit$ ，表示第 N 天的操作结束时能够获得的最大的金钱数目。答案保留3位小数。

【问题描述】

- 【输入样例】

- 3 100

- 1 1 1

- 1 2 2

- 2 2 3

- 【输出样例】

- 225.000

【问题描述】

- 【样例说明】

时间	用户操作	人民币 (元)	A券的数量	B券的数量
开户	无	100	0	0
第一天	买入100	0	50	50
第二天	卖出100%	150	0	0
第二天	买入150	0	75	37.5
第三天	卖出100%	225	0	0

【问题描述】

- 测试数据设计使得精度误差不会超过 10^{-7} 。
对于40%的测试数据，满足 $N \leq 10$ ；
对于60%的测试数据，满足 $N \leq 1\ 000$ ；
对于100%的测试数据，满足 $N \leq 100\ 000$ ；
- 对于100%的测试数据，满足：
 $0 < AK \leq 10$ ；
 $0 < BK \leq 10$ ；
 $0 < RateK \leq 100$
 $MaxProfit \leq 10$

【问题描述】

- **【提示】**

输入文件可能很大，请采用快速的读入方式。
必然存在一种最优的买卖方案满足：
每次买进操作使用完所有的人民币；
每次卖出操作卖出所有的金券。

- **【评分方法】**

本题没有部分分，你的程序的输出只有和标准答案相差不超过0.001时，才能获得该测试点的满分，否则不得分。

重要的提示

- 刚才的提示十分重要，而事实上在NOI2007考场上时，好多选手因没有看见提示而失分（当然也有牛自己推出了提示的内容）。
- 提示的意思是，要买，就花光所有钱买，要卖，就卖出所有金券。因为这样操作结果**一定**是最大的。

提示的证明

- 买进时，设1单位金券(1单位A券和R单位B券)的最大获利为 p ，一元人民币的最大获利为 q 。

- 设获利与使用人民币比例 x 的函数关系式为

$$f(x) = px + q(1 - x)$$

- 整理得

$$f(x) = (p - q)x + q$$

- 由于 $p - q$ 正负未知，则当且仅当 x 取0或1时， $f(x)$ 取到最大值，所以买进操作一定是完全的。
- 同理可证用户卖出时也一定是完全的。

算法分析

- 有了提示定理的存在，我们意识到，用户每天可能的活动有且只有以下四种
 1. 全部买进
 2. 全部卖出
 3. 全部卖出后全部买进
 4. 不进行操作

算法分析

- 第 i 天时
- 全部卖出仅当 $A[i-1] > 0$ 且 $B[i-1] > 0$ 且 $S[i-1] = 0$
- 全部买进或卖出后全部买进当 $S[i-1] > 0$ 且 $A[i-1] = 0$ 且 $B[i-1] = 0$

算法分析

- 设买进金券A数量为 x ，买进金券B数量为 y
- 按照买进规则，有以下关系

$$y = \frac{x}{R}$$

$$Va[i] * x + Vb[i] * \frac{x}{R[i]} = S[i-1]$$



$$x = \frac{S[i-1]}{Va[i] + \frac{Vb[i]}{R}}$$
$$y = \frac{x}{R}$$

算法实现

- 该题规则明显，可以用典型的搜索解决。
- 枚举用户每一天中的四种活动，然后统计出所有方案中的最优解。

搜索简析

1. void search(long k)

2. { if(k==n+1)

3. if (S[n]>Smax) Smax=S[n];

4. else{ 只卖出

5. search(k+1);

6. 只买进

7. search(k+1);

8. 卖出后买进

9. search(k+1);

10. 什么也不做

11. search(k+1);

12. }

13. }

用CmYkRgB123 Online Judge System 测评结果



CmYkRgB123
<http://www.cmykrgb123.com>

测试点	结果	运行时间(毫秒)
1	正确	1
2	正确	3
3	正确	9
4	正确	9
5	超过时间限制	1000
6	超过时间限制	1000
7	超过时间限制	1000
8	超过时间限制	1000
9	超过时间限制	1000
10	超过时间限制	1000

算法结论

- 时间复杂度 $O(2^n)$
- 编程复杂度 很低
- 代码长度 78 行
- 最后得分 40
- 一个“骗分”算法！

程序在这里



cash_search.cpp

算法分析

- 考虑用户一天三种操作
- 买入操作
显然，这种操作不会直接影响收益。
- 卖出操作
假设之前最后一次买入是第 j 天，则今天卖出有收益。今天的最大获利就是前 j 天的最大获利。
- 不操作
此时用户最大获利等于昨天的最大获利。

算法分析

- 由此我们发现，问题具有最优子策略，可以考虑使用动态规划实现。
- 状态表示
- $S[k]$ 表示用户前 k 天的最大获利
- $S[1]$ 为最初拥有人们币数量
- $S[n]$ 就是最终答案

算法实现

- 若第 k 天不进行操作, 则 $S[k]=S[k-1]$
- 若第 k 天进行买入操作, 则 $S[k]=S[k]$ (无意义)
- 若第 k 天进行卖出操作, 则 $S[k]=\max \{F[j]*\text{tran}(j, k)\}$
- j 表示在第 k 天之前最后一次进行买入操作的时间。
- $\text{tran}(j, k)$ 表示第 j 天用1元人民币买入金券在第 k 天卖出获得的人民币的数量。

算法实现

- 综上所述，我们可以得到状态转移方程
- $$S[k] = \max \left\{ \begin{array}{l} S[k-1], \\ \max \{ F[j] * \text{tran}(j, k) \} \end{array} \right\} \quad (0 < j < k)$$

tran 实现

求第*i*天用1元人民币买入金券在第*j*天卖出获得的人民币的数量，列方程，解之可得

$$x = \frac{1}{Va[i] + \frac{Vb[i]}{R[i]}}$$

$$y = \frac{x}{R[i]}$$

$$\Rightarrow \text{tran}(i, j) = x * Va[j] + y * Vb[j]$$

动态规划简析

1. void dynamic()

2. {

3. for (i=2; i<=n; i++)

4. {

5. tMax=0;

6. for (j=1; j<=i-1; j++)

7. tMax=max(tMax, S[j]*tran(j, i));

8. S[i]=max(S[i-1], tMax);

9. }

10. }

用CmYkRgB123 Online Judge System 测评结果



CmYkRgB123

<http://www.cmykrgb123.com>

测试点	结果	运行时间(毫秒)
1	正确	2
2	正确	2
3	正确	2
4	正确	2
5	正确	16
6	正确	49
7	超过时间限制	1000
8	超过时间限制	1000
9	超过时间限制	1000
10	超过时间限制	1000

算法结论

- 时间复杂度 $O(n^2)$
- 编程复杂度 较低
- 代码长度 62 行
- 最后得分 60
- 一个可行算法!

程序在这里



Cash_dp.cpp

进一步优化

- 难度加大，在此不再进行讨论。
- 有兴趣可以参考
 - 《NOI 2007 总结与解题》余林韵
(运用平衡二叉排序树Treap的方法优化)
 - 《〈货币兑换〉解题报告》郭华阳
(运用线性规划的方法优化)

调兵遣将

CmYkRgB123

【 题目描述 】

- 我军截获的情报显示，敌军正在集结兵力试图向我军重要的军械研究所发起进攻。由于我军正处于多线作战的状态，无法抽调大批兵力前去支援，指挥部决定通过有效的战前部署来提高胜率，减少伤亡和损失。
该军械研究所的平面图可以看作是一个 $N \times M$ 的矩阵，每个 1×1 的格子都表示一个区域，每个区域只与它上下左右的四个区域相邻。每个区域的用途可分为以下3种之一：

【 题目描述 】

1. 该区域被用于军事研究（用字母' 0' 表示）
2. 该区域内驻扎有一个机械化中队（用' #' 表示）；
3. 该区域是空地（用' .' 表示）。
 - 由于空间有限，任一个 $1*1$ 的格子内都无法驻扎两队以上的机械化中队（包括两队），否则会大大降低战斗时的机动性。

【 题目描述 】

- 遗憾的是，由于战前估计不足，我军的防御部署显得十分分散，这很容易让敌军所擅长的偷袭战术得逞。为了确保万无一失，我军决定利用为数不多的防御部队以最少的移动步骤将所有重要研究区域都包围起来。所谓的“包围”即从该矩阵边界侵入的敌军找不到任意一条路，使得他们不遭受任何机械化中队的反抗就能到达某研究区域。

【 题目描述 】

- 由于军队内部的传令权限的限制，每个单位时间指挥部只能向所有中队中的一个中队下达指令（朝上/下/左/右移动1格）。由于时间紧迫，指挥部希望能够尽快完成部署，这个任务就交给你来完成。

注意：在部署的过程中军队可以进入研究区域，而在最终的部署结果中军队不可以在研究区域中。另外，在任何时刻，两个军队都不可以在同一个方格中。

【 题目描述 】

- 【输入格式】

该题为提交答案型试题，所有输入数据
surround1.in~surround10.in在考试开始前已
被存入各位选手的试题目录下。

对于每个数据：

第一行包括2个整数N，M，接下来N行，每行包
括M个字符（‘.’，‘ ’，‘0’或‘#’）

【 题目描述 】

- 【输出格式】
- 针对给定的10个输入文件
surround1.in~surround10.in，你需要分别提交你的输出文件
surround1.out~sourround10.out。
每个输出文件的第一行，包括你的答案所花费的时间T
- 接下来T行，按顺序输出每条命令，每行包括4个整数x1, y1, x2, y2，表示将位于(x1, y1)的部队移向(x2, y2)。

【 题目描述 】

- 【输入样例】

- 5 5
..##.
#...#
#000#
#..0#
.###.

- 【输出样例】

- 1
2 1 2 2

问题分析

- 该题是个提交答案型试题。
- 做提交答案型试题要遵循“看数据，找规律”六字原则。

数据规律

```

surround1.in - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

100 100
####.
#00.
#00.
#.#.
#####.
#.#00.
#.#00.
#.#00.
#.#00.
#.#00.
#####.
#####.
#.#0000#.
#.#0000#.
#####0000#.
#.#0000#.
#.#0000#.
#####0000#.#.
#.#0000#.#.#.
#.#0000#.#.#.
#####.#.#.
#.#0.#.

```

数据规律

```
surround5.in - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

.....
.###.
.0.#.
.0.#.
#00000#
#.#.
#.#.
.###.#####
#.#.0
#.#.0
#.#.0
#.#.0
#.#.0
#.#.0
#00000000000#
#.#.
#.#.
#.#.
#.#.
#.#.
#.#.
#####
.....
```

```
surround3.in - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

3000 00
.....
#.#.
#.#.
0#00#0
0.#.00.#.0
0.#.00.#.0
0.#.00.#.0
0.#.00.#.0
#####0#####
0.#.00.#.0
0.#.00.#.0
0.#.00.#.0
0#00#0
#.#.
#.#.
.....
```

数据规律

- 第1, 2 个测试数据为手工测试点, 看着找吧。
- 第3, 4, 5, 8为较特殊的数据。4中0十分分散, 并且有规律。可以用最短路做。
- 3, 5, 8都是模板型数据, 即: 处理一种小数据然后复制若干遍。
- 剩余的6, 7, 9, 10为随机大数据。将外围(边沿)全部覆盖就是一种好方法。

参考算法

- 网络流+最佳匹配。首先运用最小割求出最少需要放置
- # 的位置，其次计算出哪些# 是可移动的。最后运用最佳匹配，将可移动
- 的# 作为X 部，需放置的地方作为Y 部，两者间距离作为权值，求出答案。
- 更改结点间在网络流中的容量，多次运行，取最优解。

总结

- 写这次解题报告，我学习了不少。学习了不仅仅是各种算法，更是一种能力，一种思维，一种心态——分析、抽象与解决问题的能力，计算的思维 (Thinking in Computing)，能够处变不惊、冷静分析的心态。
- 同时，这是我第一次了解NOI难度的题，相信NOI已经就在不远的前方。
- Let's Struggle Together!

参考资料

- 《NOI 2007 总结与解题》 余林韵
- 《2007 全国信息学奥林匹克年鉴》 王宏
- 《计算机算法设计与分析》 王晓东
- 《算法艺术与信息学竞赛》 刘汝佳 黄亮
- 《Data Structures and Algorithm 》
Mark Allen Weiss

感谢

1. 感谢生养我的父母
2. 感谢常庆卫老师
3. 感谢杨晓宇、王寅同学
4. 感谢高翔、李垚、马浩同学
5. 感谢秦臻同学



CmYkRgB123

<http://www.cmykrgb123.com>