**★★★★★**

**題組：Problem Set Archive** **with Online Judge**

**題號：10615 : Rooks (1)**

**解題者：陳冠穎**

**解題日期：2025年5月29日**

**題意：**

給定一個N\*N的棋盤，輸入中’.’代表空棋格，’\*’代表一個rook。要求為每一個rook上色，並且同一行和同一列不可以有相同顏色。要求輸出最少上色所需顏色和上色情況，用數字代表顏色(0代表空棋格)。(多種輸出情況皆能接受)

**題意範例：**

輸入:

4

\*.\*.

\*.\*.

\*\*\*.

..\*\*

對應輸出:

4

1 0 2 0

3 0 1 0

2 1 3 0

0 0 4 1

**解法：**

將這個問題可以轉化為bipartite graph(二分圖)著色問題。每個節點不能有相同顏色的邊。節點是行和列的二分圖，連結的邊代表每個棋子。

(1)找出給定資料中節點的最大邊數，決定之後的上色次數n

(2)創造補邊，為了使之後上色時，每個節點都有n條邊可以上色(不會沒有配對)

(3)用dfs尋找row0~row3的最大配對，找完將真實邊上色，並將所有配對邊刪除

(4)重複(3) n次

(5)輸出儲存結果

**解法範例：**

無

**討論：**

無

**程式：**

#include <algorithm>

#include <iostream>

#include <cstring>

#include <cstdio>

#include <cmath>

#include <vector>

#include <stack>

#include <queue>

using namespace std;

const int maxn = 100 + 3;

bool real[maxn][maxn], s[maxn], t[maxn];

int n, inrow[maxn], incow[maxn], match[maxn], col[maxn][maxn], w[maxn][maxn];

bool dfs(int u) {

 s[u] = true;

 for(int v = 0; v < n; ++v)

 if(w[u][v] > 0 && !t[v]) {

 t[v] = true;

 if(match[v] == -1 || dfs(match[v])) {

 match[v] = u;

 return true;

 }

 }

 return false;

}

int bipartite() {

 int cnt = 0;

 fill(match, match + n, -1);

 for(int i = 0; i < n; ++i) {

 fill(s, s + n, false);

 fill(t, t + n, false);

 cnt += dfs(i);

 }

 return cnt;

}

inline void relax(int &a, int b) {

 if(b > a) a = b;

}

inline int ReadInt() {

 int n = 0, ch = getchar();

 while(ch < '0' || ch > '9') ch = getchar();

 do {

 n = n \* 10 + ch - '0';

 ch = getchar();

 }while(ch >= '0' && ch <= '9');

 return n;

}

int main() {

 int caseNum = ReadInt();

 while(caseNum--) {

 n = ReadInt();

 fill(inrow, inrow + n, 0);

 fill(incow, incow + n, 0);

 for(int i = 0; i < n; ++i) {

 for(int j = 0; j < n; ++j) {

 w[i][j] = real[i][j] = getchar() == '\*';

 if(real[i][j]) inrow[i]++, incow[j]++;

 }

 while(getchar() == ' ');

 }

 int ans = 0;

 for(int i = 0; i < n; ++i) relax(ans, max(inrow[i], incow[i]));

 for(int i = 0; i < n; ++i)

 if(inrow[i] < ans)

 for(int j = 0; j < n; ++j) {

 while(inrow[i] < ans && incow[j] < ans) {

 inrow[i]++; incow[j]++;

 w[i][j]++;

 }

 }

 for(int i = 1; i <= ans; ++i) {

 bipartite();

 for(int j = 0; j < n; ++j) {

 if(match[j] == -1) continue;

 if(real[match[j]][j]) col[match[j]][j] = i;

 w[match[j]][j]--;

 }

 }

 printf("%d\n", ans);

 for(int i = 0; i < n; ++i)

 for(int j = 0; j < n; ++j)

 printf("%d%c", real[i][j] ? col[i][j] : 0, j + 1 == n ? '\n' : ' ');

 }

 return 0;

}