Chapter 1

图论

1.1 最大团

```
/*
最大团搜索算法
Si = {vi, vi+1, ... , vn}
mc[i] = MC(Si) = 点集合 Si 的最大团
MC(V) = mc[1]
mc[i] = mc[i+1] || mc[i] = mc[i+1] + 1 因为最多增加一个点
后一种情况发生是在 Si 中找到一个包含vi的团
所以只要搜 是否在 Si 中有一个包含vi且比当前最大团还大的团
*/
#include <bits/stdc++.h>
#define rep(i, l, r) for (int (i) = (l); (i) <= (r); (i)
    ++)
#define log(x) cout << #x << " = " << x << endl
#define mem(x, y) memset((x), (y), sizeof(x))
using namespace std;
int n, ans;
bool found;
const int N = 50 + 9;
int g[N][N];
int len[N], mc[N], li[N][N];
void dfs(int sz)
{
    if (len[sz] == 0) {
        if (sz > ans) {
            ans = sz;
            found = true;
        }
    }
```

```
28      for (int k = 0; k < len[sz] && !found; k++) {
29          if (sz + len[sz] - k <= ans) break;
30          int i = li[sz][k];
31          if (sz + mc[i] <= ans) break;
32          len[sz+1] = 0;
33          for (int j = k + 1; j < len[sz]; j++) {
34              if (g[i][li[sz][j]])
35                  li[sz+1][len[sz+1]++] = li[sz][j];
36          }
37          dfs(sz+1);
38      }
39  }
40  int max_cluster()
41  {
42      mc[n] = ans = 1;
43      for (int i = n - 1; i; i--) {
44          found = false;
45          len[1] = 0;
46          rep(j, i+1, n) {
47              if (g[i][j]) li[1][len[1]++] = j;
48          }
49          dfs(1);
50          mc[i] = ans;
51      }
52      return ans;
53  }
54  int solve()
55  {
56      rep(i, 1, n) rep(j, 1, n) scanf("%d", &g[i][j]);
57      return max_cluster();
58  }
59  int main()
60  {
61      for (;;) {
62          scanf("%d", &n);
63          if (n == 0) break;
64          printf("%d\n", solve());
65      }
66  }
```