```
const int MAXN = 210;
const int MAXM = 500010;
const int inf = 2E9;

typedef struct {int v,next,val;} edge;
struct SAP {
    edge e[MAXM];
    int p[MAXN],eid;
    inline void clear(){memset(p,-1,sizeof(p));eid=0;}
    inline void insert1(int from,int to,int val) {
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
        swap(from,to);
        e[eid].v=to;
        e[eid].val=0;
        e[eid].next=p[from];
        p[from]=eid++;
    }
    inline void insert2(int from,int to,int val) {
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
        swap(from,to);
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
    }
    int n;//为点数n 为边数m
    int h[MAXN];
    int gap[MAXN];
    int source,sink;
    inline int dfs(int pos,int cost) {
        if (pos==sink) {
            return cost;
        }
        int j,minh=n-1,lv=cost,d;
        for (j=p[pos];j!=-1;j=e[j].next) {
            int v=e[j].v,val=e[j].val;
            if(val>0) {
                if (h[v]+1==h[pos]) {
                    if (lv<e[j].val) d=lv;
                    else d=e[j].val;
                    d=dfs(v,d);
                    e[j].val-=d;
                    e[j^1].val+=d;
                    lv-=d;
                    if (h[source]>=n) return cost-lv;
                    if (lv==0) break;
```

```
53                  }
54                  if (h[v]<minh)  minh=h[v];
55              }
56          }
57          if (lv==cost) {
58              --gap[h[pos]];
59              if (gap[h[pos]]==0) h[source]=n;
60              h[pos]=minh+1;
61              ++gap[h[pos]];
62          }
63          return cost-lv;
64      }
65      void read(int ll[MAXN][MAXN],int S,int T,int N) {
66          clear();
67          source = S; sink = T; n = N;
68          for (int i = 0; i <= N; i++) {
69              for (int j = 0; j <= N; j++) {
70                  if (ll[i][j]) {
71                      insert1(i,j,ll[i][j]);
72                  }
73              }
74          }
75      }
76      int run() {
77          int ret=0;
78          memset(gap,0,sizeof(gap));
79          memset(h,0,sizeof(h));
80          gap[source]=n;
81          while (h[source]<n) ret+=dfs(source,inf);
82          return ret;
83      }
84  } solver;
```