// 强连通分量

int dfs\_clock = 0, pre[MaxN],low[MaxN];

int scc\_cnt, sccno[MaxN], size[MaxN];

stack<int> S; vector<int> G[MaxN];

void Add(int a,int b) {G[a].push\_back(b);}

void dfs(int u) {

pre[u] = low[u] = ++dfs\_clock;

S.push(u);

for (int i = 0; i < G[u].size(); i++) {

int v = G[u][i];

if (!pre[v]) {

dfs(v);

low[u] = min(low[u], low[v]);

}

else if (!sccno[v])

low[u] = min(low[u],pre[v]);

}

if (low[u] == pre[u]) {

scc\_cnt++;

int original\_size = S.size(), tmp = 0;

do {

tmp = S.top(); S.pop();

sccno[tmp] = scc\_cnt;

} while (tmp != u);

size[scc\_cnt] = original\_size - S.size();

} } // end of dfs

int main()

{ for (int i=0; i<n; i++) if (!pre[i]) dfs(i); }