

```

1  #include <stdio>
2  #include <string>
3  #include <stdlib>
4  #include <math>
5  #include <ctime>
6  #include <cassert>
7  const int MAXN = 510;
8
9  int N,M;
10 int adj[MAXN][MAXN];
11
12 bool visit[MAXN];
13 int dist[MAXN];
14 int vec[MAXN];
15 int work() {
16     if (N == 2) return adj[vec[0]][vec[1]];
17     memset(visit,0,sizeof(visit));
18     visit[vec[0]] = true; int cnt = 1;
19     for (int i = 0; i < N; i++) {
20         dist[vec[i]] = adj[vec[0]][vec[i]];
21     }
22     int p1 = 0; int p2 = 0;
23     for (int nn = N - 1; nn ; nn --) {
24         int pos = vec[0];
25         for (int i = 0; i < N; i++) {
26             if (visit[vec[i]]) continue;
27             if (dist[vec[i]] > dist[pos]) pos = vec[i];
28         }
29         p2 = p1; p1 = pos;
30         visit[pos] = true; cnt ++;
31         for (int i = 0; i < N; i++) {
32             if (visit[vec[i]]) continue;
33             dist[vec[i]] += adj[pos][vec[i]];
34         }
35     }
36     if (cnt < N) return 0;
37     int ans = dist[p1];
38     for (int i = 0; i < N; i++) {
39         if (adj[vec[i]][p1] && vec[i] != p2) {
40             adj[vec[i]][p2] += adj[vec[i]][p1];
41             adj[p2][vec[i]] += adj[p1][vec[i]];
42         }
43     }
44     for (int i = 0; i < N; i++) {
45         if (vec[i] == p1) {
46             vec[i] = vec[--N];
47             break;
48         }
49     }
50     int tmp = work();
51     return tmp > ans ? ans : tmp;
52 }

```