

迷阵突围

code by 刘业圣

Code

<https://pastebin.skymoon.top/?7efe5e9479fe562d#6SPiPCz4oYk4dBaMpESbpQPwwUwL2u1rtkgn752je7jd>

思路

1. 记录最短路径（记录路径点/记录路径边）
2. 切断边，再次Dijkstra

枚举两个顶点之间最短路上的每条边，每次在去掉这条边的剩下的图中计算最短路，取其中最小的一个答案就是最终次短路的答案

数据存储

邻接表

```
struct edge {
    int v;
    double w;
    int next, f;
};

struct edge e[40000];
```

```
int p[MAX]
```

读数据

```
memset(p, -1, sizeof(p));
int n, m;
scanf("%d%d", &n, &m);
for (int i = 1; i ≤ n; i++) {
    scanf("%d%d", &coordinatex[i], &coordinatey[i]);
}

for (int i = 1; i ≤ m; i++) {
    int p, q;
    scanf("%d%d", &p, &q);
    double w = distance(p, q);
    insert2(p, q, w);
}
```

```
void insert(int u, int v, double w) {
    struct edge t = {v, w, p[u], 1};
    e[eid] = t;
    p[u] = eid++;
}

void insert2(int u, int v, double w) {
    insert(u, v, w);
    insert(v, u, w);
}
```

```
double distance(int p, int q) {
    return sqrt(
        (coordinatex[p] - coordinatex[q]) * (coordinatex[p] - coordinatex[q]) +
        (coordinatey[p] - coordinatey[q]) * (coordinatey[p] - coordinatey[q]));
}
```

Dijkstra

```
void dijkstra(int n) {
    memset(visit, 0, sizeof(visit));
    for (int i = 1; i ≤ n; i++) {
        d[i] = inf;
    }
    d[1] = 0;

    for (int i = 1; i ≤ n; i++) {
        double mind = inf;
        int u = 0;
        for (int j = 1; j ≤ n; j++) {
            if (!visit[j] && d[j] < mind) {
                mind = d[j];
                u = j;
            }
        }
        if (mind == inf) {
            break;
        }
        visit[u] = 1;
        for (int j = p[u]; j ≠ -1; j = e[j].next) {
            if (e[j].f) {
                int v = e[j].v;
                if (d[u] + e[j].w < d[v]) {
                    d[v] = d[u] + e[j].w;

                    last[v] = j;
                }
            }
        }
    }
}
```

第一次Dijkstra

```
memset(last, 0, sizeof(last));
dijkstra1(n);
```

第二次

```
int id = last[n];
last[1] = -1;

while (id  $\neq$  -1) {
    e[id].f = e[id ^ 1].f = 0;
    dijkstra1(n);
    if (d[n] < subshortest) {
        subshortest = d[n];
    }
    e[id].f = e[id ^ 1].f = 1;
    id = last[e[id ^ 1].v];
}
```