

# E5E 上班

## 题目描述

观赛完运动会之后，Zhoues 需要立马回去钢厂上班，由于北航到钢厂的路径有很多条，他希望选择一条最短的！

但是貌似最短的路径有很多条，Zhoues 想知道有多少条最短的路径呢？

形式化来说，给一个  $n$  个点  $m$  条边的无向图，每条边的权值均为 1，求从点  $s$  到点  $t$  的最短路数量，两条最短路不同当且仅当包含了不同的边。由于答案可能很大，你只需要输出答案对 998 244 353 取模后的结果。

## 思路一

求单源最短路？Dijkstra 大法好。

只需要在 Dijkstra 遍历的时候连带判断最短路条数即可，时间复杂度  $O(m \log n)$

```
#define MK make_pair
#define re register
#define PII pair<int,int>
priority_queue<PII,vector<PII>,greater<PII>> q;

// dist 数组储存距离起点的距离
// nums 数组储存到达该点的最短路条数

void Dijkstra(int begin) {
    dist[begin] = 0;    nums[begin] = 1;
    q.push(MK(dist[begin], begin));    //起点加入队列
    while(!q.empty()) {
        int top = q.top().second;    q.pop();    //取出队首元素
        if(flag[top]) continue;
        flag[top] = true;
        for(re int i = head[top]; i != 0; i = edge[i].nex) {
            int beh = edge[i].to;
            if(dist[beh] == dist[top] + 1) {    // 如果到达的也是最短路
                nums[beh] = (nums[beh] + nums[top]) % MOD;
                continue;
            }
            if(dist[beh] > dist[top] + 1) { //更新最短路
                dist[beh] = dist[top] + 1;
                nums[beh] = nums[top];
                q.push(MK(dist[beh], beh));
            }
        }
    }
}
```

## 思路二

观察到每条边的权值均为1，emmmmmm?? 好东西!! 尝试考虑暴力 BFS 搜索。



原因是：

- BFS 本质就是边权为 1 的最短路。
- 不妨看一下代码结构，BFS 是用的 queue，Dij 是用的 priority\_queue。
- 当边权是 1 时，做 Dij 时新松弛得到的长度肯定不小于队列里的所有长度，也就是直接接队尾。用优先队列没有必要。

此时时间复杂度为  $O(m)$

```
// dist 数组储存距离起点的距离
// nums 数组储存到达该点的最短路条数

void BFS(int begin) {
    dist[begin] = 1;    nums[begin] = 1;
    q.push(begin);
    while(!q.empty()) {
        int top = q.front();    q.pop();
        for(re int i = head[top]; i != 0; i = edge[i].nex) {
            int beh = edge[i].to;
            if(dist[beh] == 0 || dist[beh] == dist[top] + 1) {
                if(dist[beh] == 0) q.push(beh);
                nums[beh] = (nums[beh] + nums[top]) % MOD;
                dist[beh] = dist[top] + 1;
            }
        }
    }
}
```