

## E5-E 题

### 题目：

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。

无权图（权值均为 1 的图）的最短路，可以用 bfs 来求。到  $i$  点最短路的长度，即为  $i$  点的深度，记为  $deep[i]$ 。需要维护一计数器  $cnt[i]$  记录最短路的方案数。先将起点放入队列，然后开始从队列中取数，对于每个从队列中取出的数  $t_0$ ，以  $t_0$  为端点的边的另一端点  $j$ ，如果没有被确定最短路，就将其放入队列，同时更新  $cnt$ 。

因为  $cnt[t_0]$  一定是上一层的点，以确定了  $deep$  和  $cnt$ ，有多少条路从起点到  $t_0$ ，就有多少条路从起点经过  $t_0$  到  $j$ 。而与  $t_0$  同深度、在  $t_0$  之前被遍历的点没有与  $j$  连通的点时， $cnt[j]$  为 0，此时  $cnt[j]=cnt[t_0]$ ，即目前为止从起点只能通过  $t_0$  到  $j$ ；与  $t_0$  同深度、在  $t_0$  之前被遍历的点有与  $j$  连通的点时，从起点到  $j$  的最短路数量分为经过  $j$  的（ $cnt[t_0]$ ）和不经过  $t_0$  的（ $cnt[j]$ ），此时  $cnt[j]=cnt[j]+cnt[t_0]$ 。故  $cnt[j]=cnt[j]+cnt[t_0]$ 。

关键代码如下：

```
q.push(s);
deep[s] = 1;
cnt[s] = 1; // 这里设置第一个点的最短路条数
while (q.size())
{
    int t0 = q.front();
    q.pop();
    if (st[t0]) continue; // 每个点可能多次被遍历，所以可能多次被加在队列中
    st[t0] = true; // 标记已经遍历过
    for (int i = h[t0]; i != -1; i = ne[i]) { // h[t]为t为端点的第一条边的位置，ne[i]是i为端点的下一条边
        int j = e[i]; // e[i]表示该条以i为一端点的边的另一端点
        if (!st[j] && deep[j] != deep[t0]) { // 如果该点没有被确定最短路，并且不与根节点在同一层中，就把这些点执行更新，并把层数加一，然后放进队列
            cnt[j] = (cnt[j] + cnt[t0]) % mod; // 每一点的最短路，一直会更新
            deep[j] = deep[t0] + 1; // 层数加1
            q.push(j); // 放入队列
        }
    }
}
if (t0 == tt) return; // 到了题目要求的终点就可以返回了
}
```