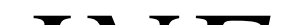
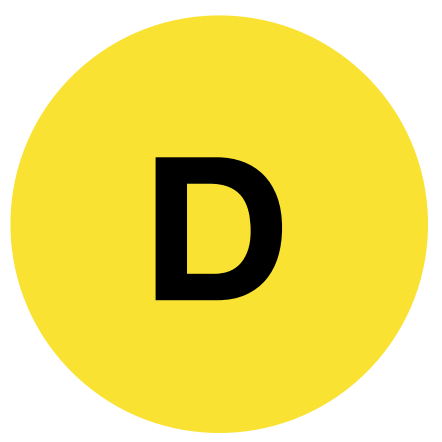
```
void Dijkstra(int start)
{
    init();
    dist[start] = 0;
    while(true){
        int min = INT_MAX;
        int min_index = -1;
        for( int i = 1; i <= n; i++){
            if(min > dist[i] && !collected[i]){
                min = dist[i];
                min_index = i;
            }
        }           // 14-19行为找出最近的未被收录的点
        if(min_index == -1){    // 如果找不到，跳出循环
            break;
        }
        collected[min_index] = true;    // 将该点收录，
        for( int i = 1; i <= n; i++){   // 遍历该点的邻接点
            if(map[min_index][i] != -1 && !collected[i]){
                int temp = dist[min_index] + map[min_index][i];
                if(temp < dist[i]){
                    dist[i] = temp;
                    path[i] = min_index;
                }
            }
        }
    }
}
```

6

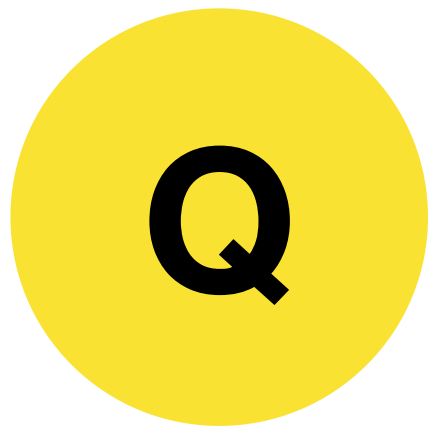

INF

-1


D


Y


T


Q


P

10

3

5

2

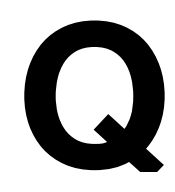
8

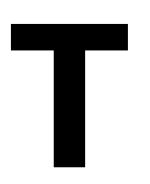
9


dist

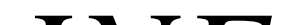
path

collected





0


INF


INF

-1

-1

-1

-1

false


false

false

false

false

true


INF

$\texttt{min} = 0$

$\texttt{min\_index} = P$

找到与起始点最近且未被收录的点 P

将点 P 收录到集合中


P