

```

/* 邻接表存储 - 无权图的单源最短路算法 */

/* dist[]和path[]全部初始化为-1 */
void Unweighted ( LGraph Graph, int dist[], int path[], Vertex S )
{
    Queue Q;
    Vertex V;
    PtrToAdjVNode W;

    Q = CreateQueue( Graph->Nv ); /* 创建空队列, MaxSize为外部定义的常数 */
    dist[S] = 0; /* 初始化源点 */
    AddQ (Q, S);

    while( !IsEmpty(Q) ){
        V = DeleteQ(Q);
        for ( W=Graph->G[V].FirstEdge; W; W=W->Next ) /* 对V的每个邻接点W->AdjV */
            if ( dist[W->AdjV]==-1 ) { /* 若W->AdjV未被访问过 */
                dist[W->AdjV] = dist[V]+1; /* W->AdjV到S的距离更新 */
                path[W->AdjV] = V; /* 将V记录在S到W->AdjV的路径上 */
                AddQ(Q, W->AdjV);
            }
    } /* while结束*/
}

```