

无权单源最短路径算法，求顶点  $v$  到其他各顶点的最短路径

```
void Graph_List::ShortestPath(const int v)
{
    int u, k;
    Edge *p;
    int n = graphsize;
    int* path = new int[graphsize];
    int* dist = new int[graphsize];
    Queue<int> Q; // 声明一个队列
    for(int i = 0; i < n; i++) // 数组 path 和 dist 初始化
        {path[i] = -1; dist[i] = -1; }
    dist[v] = 0; // 初始顶点 v 的 dist 值为 0
    Q.QInsert(v); // 初始顶点 v 入队
    while(!Q.QEmpty())
    {
        u = Q.QDelete(); // 删除队头顶点 u
        p = Head[u].adjacent; // p 为 u 的边链表的头指针
        // 将 u 的未访问的邻接顶点入队，并修改其 path[] 值和 dist[] 值
        while(p != NULL)
        {
            k = p->VerAdj;
            if( dist[k] == -1 )
            {
                Q.QInsert(k);
                dist[k] = dist[u] + 1;
                path[k] = u;
            }
            p = p->link;
        }
    }
    for (i = 0; i < n; i++) cout<<path[i]<<" ";
    for (i = 0; i < n; i++) cout<<dist[i]<<" ";
    delete[] path;
    delete[] dist;
}
```