数学归纳法证明Dijkstra算法

Dijkstra算法：

有一个非负边的图，令集合`S`表示已经确定到源点最短路的点的集合，令集合`T`表示还未确定的点的集合，令`dist[i]`表示从源点到顶点`i`的距离，初始的时候`dist[i]`中的值是无穷，所有的点都在集合`T`中，`w[x, i]`表示点`x -> i`的权值，我们重复以下步骤：

1. 在集合`T`中找到离源点路径最短的点，也就是`T`中的`dist[i]`最小的点`x`
2. 将这个点`x`放入集合`S`中，并利用这个点更新`T`中的其他的点到源点的距离，即`dist[i] = min(dist[i], dist[x] + w[x, i])`
3. 重复以上两个步骤，一直到最后一个点的距离更新完毕

证明：

我们需要证明的就是每次从集合`T`中挑出来的点到源点的距离必然是算法结束后的最短距离

首先，第一次加入`S`的点`x0`，其距离`dist[x0] = 0`一定是最短距离，即我们第一次将源点加入到集合，然后更新其余点的距离

然后，假设集合`T`中离源点路径最短的点，也就是`T`中的`dist[i]`最小的点`xk`，此时就是点`xk`到源点的最短距离，然后更新其余点的距离

最后，我们来考察从集合`T`中挑出的下一个`dist[i]`最小的点`xm`，此时这个点的距离为`dist[xm]`，而这个`dist[xm]`就是点`xm`到源点距离最近的距离，我们用反证法来证明：

假设此时的`dist[xm]`不是点`xm`到源点的最短距离，那么必然存在另外一条路线`L`到到源点，其数值为`dist[xm']`

我们注意到这样两个事实：

集合`T`中的点，只要距离不为无穷的点，这个点对应的距离`dist[i]`对应的路线中这个点的前驱点必然是在集合`S`中

> 如果这个点的`dist[i]`存在一个数值，那么这个数值必然对应着一条路线到源点，而这个数值根据算法一定是由集合`S`中的点更新得来的，所以其前驱必然在集合`S`中

集合`T`中的点，只要距离不为无穷的点，这个点对应的距离`dist[i]`对应的路线一定是源点经过集合`S`中的点到达这个点的距离最短的路线

> 因为算法每加入一个新的点进入集合`S`中就必然更新所有点的距离，所以目前集合`T`中的点此时的路线一定是源点经过集合`S`到达的最短的

所以根据路线`L`，`xm`点的前驱必然在集合`T`中，假设此时的路线为`L'`，而`L'`这条路线也是先从集合`S`再到集合`T`的，`L' = x0 -> x1 -> x2....->x(m-1) ->x(m)`，假设点`xk`是`L'`路线上第一个在集合`T`中的点即`x(k - 1) -> xk`，其中`x(k - 1)`在集合`S`中，`xk`在集合`T`中，那么就有：`dist[xk] + w[xk, xm] < dist[xm]`，由于边权非负，所以有`dist[xk] < dist[xm]`，这与`dist[xm]`是集合`T`中最小的矛盾！所以此时的`dist[xm]`就是点`xm`到源点的最短距离，证毕。