名词解释：

若图graph含有连通的A、B两点，则图中的有向边有两条，它们互为反向有向边；有向边的起点称为from，终点称为to，边的长度称为d。算法描述中，set表示有向边集合，queue表示有向边队列。每个节点存储一个shortest值（代表从开始节点到本节点的最短距离），和一个previous指针（表示从开始节点到本节点的有向路径上，本节点的前一个节点）。

1. 给定graph和起始节点s，每个点的shortest均置为0
2. 将以s为to的有向边放入set
3. 将以s为from的有向边放入queue的尾部
4. （main loop）从queue头部取出一条有向边e，若e不为null，执行5，否则结束循环
5. 计算e.d+e.from.shortest的值v，若e.to.shortest为0或大于v，则将e.to.shortest置为v，将e.to.previous指向e.from（每次改变previous指针的指向时，执行6），否则do nothing。然后，对于每一条“以e.to为from的有向边”（e的反向有向边除外，即不走回头路），均做如下操作：将其放入set，若放入成功，将其再放入queue的尾部，否则（此有向边之前已经计算过，不再计算第二遍）do nothing。继续执行4
6. 遍历当前节点c的各个邻居节点n，若n的previous为c，则将n的shortest值置为v=c.shortest+c与n间的距离。然后对n递归进行本步骤

结束循环时，graph中的任何一个节点，只要与s连通，都能沿着previous指针回溯到s。

注：在queue的尾部放入，头部取出，保证了以s往外的发散按照广度优先遍历进行。