

4-13:

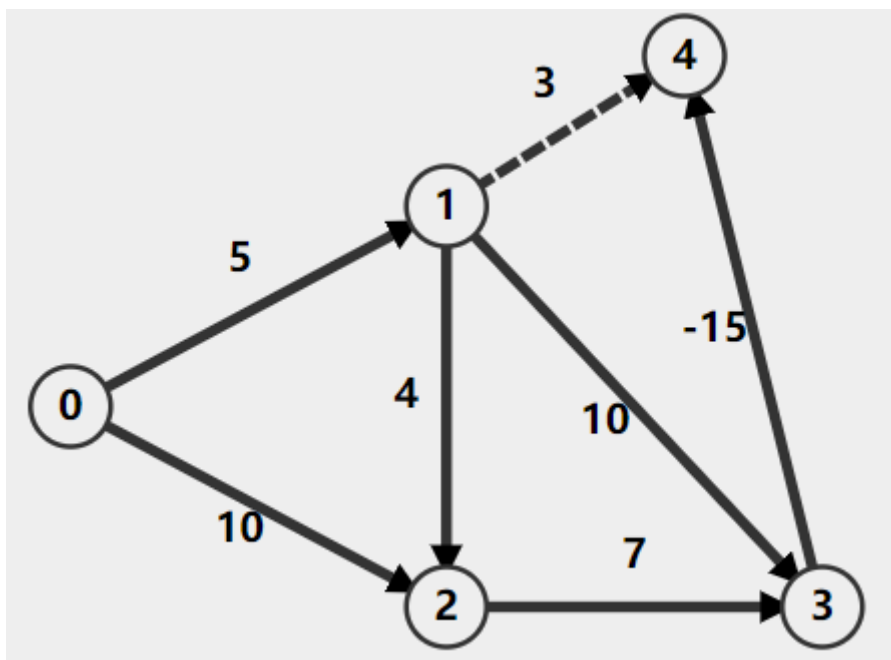
试举例说明如果允许带权有向图中某些边的权为负实数，则 *Dijkstra* 算法不能正确求得从源到所有其他顶点的最短路径长度。

题目分析

Dijkstra 算法的核心在于每次选取与已选取结点集合 S 最短的结点，同时更新所有该结点的相邻结点与集合 S 中结点的最短距离，我们每次只根据集合内点的临边来更新，只要目标点加入集合，则算法结束，得到“最短路径长度”。但这个算法的基于前提是贪心思想，也就是不考虑“把没在 S 集合内的点加入集合”，因为只可能会增加无用的边（对于正权图而言，路径长显然是递增的，即 $v[i+1] > v[i]$ ）若带权有向图中某些边的权值为负实数，*Dijkstra* 算法就会因此失效，其原因就出在没有考虑所有结点。

典型反例

如图是一个带权有向图：



设 $v[i]$ 表示第 i 个结点到 v_0 （起始结点）的最短距离。对于该图，我们令起始结点为结点0，终止结点为结点4。则在 *Dijkstra* 算法中 *while* 循环每次进行后的结果如下：

结点	$V[0]$	$V[1]$	$V[2]$	$V[3]$	$V[4]$
结点0开始	0	5	10	∞	∞
选取结点1	0	5	9	10	8
选取结点4	0	5	9	10	8

由于结点4加入 S 集合，因此算法结束，通过 *Dijkstra* 求出的最短路径为 $v[4] = 8$ ，但显然，对于该负权图是不成立的，因为存在一条更短的路径：

$$0 \rightarrow 3 \rightarrow 4$$

(1)

对于该路径，权值总和为 0，显然比 *Dijkstra* 算法更优，说明 *Dijkstra* 算法不能正确求得从源到所有其他顶点的最短路径长度。