Bellman-Ford算法

主讲人：邓哲也

大纲

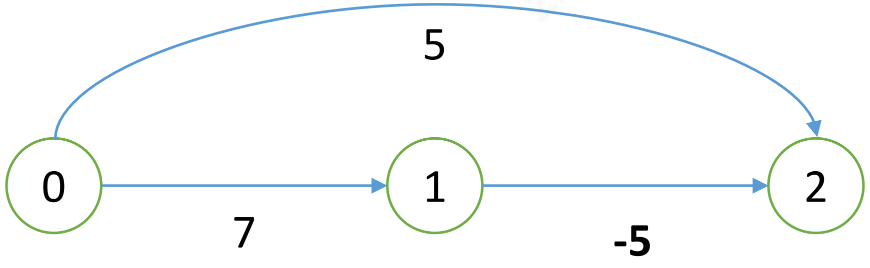
- Dijkstra算法的局限性
- Bellman-Ford算法思想
- Bellman-Ford算法时间复杂度
- Bellman-Ford算法判断负环
- Bellman-Ford算法与Dijkstra算法比较

Dijkstra算法的局限性

- 之前我们介绍的 Dijkstra 算法，要求图中各边上的权值大于或等于0.
- 如果图中有负边权的边，会发生什么呢？

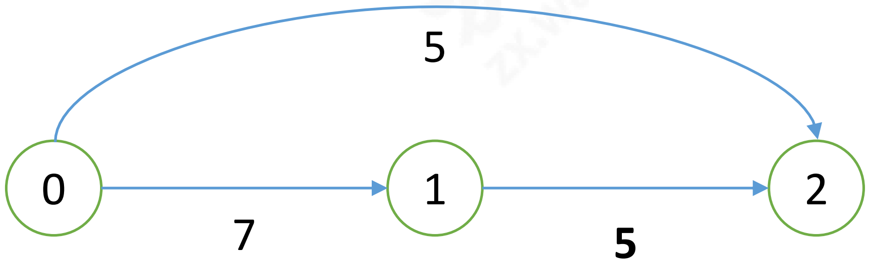
Dijkstra算法的局限性

- 用 Dijkstra算法求得 0 到 2 的最短路是 dist[2]，即 0 到 2 的直接路径，长度为 5。
- 但从 0 到 2 的最短路径应该是 0 -> 1 -> 2，长度为 2.



Dijkstra算法的局限性

- 如果把边 <1,2> 的权值由 -5 改为 5，则采用 Dijkstra 算法得到的结果是正确的。



Dijkstra算法的局限性

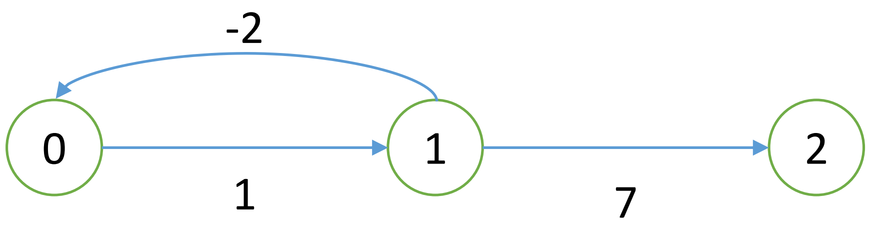
- 为什么图中出现负权值边时，dijkstra 算法求解得到的最短路径就是错的了？
- 因为 dijkstra 算法在利用顶点 u 的 dist[] 值去递推 T 集合各顶点的 dist[k] 时，前提是顶点 u 的 dist[] 值是当前 T 集合中最短路径长度最小的。
- 如果边权都是非负的，那么是没有问题的。
- 如果边权有负的，如同上一个例子，这个前提就是错误的了。

Bellman-Ford算法思想

- 为了能够求解边权为负的单源最短路径问题，Bellman 和 Ford 提出了从源点逐次经过其他顶点，以缩短到达终点的最短路径长度的方法。
- 该方法也有一个限制条件：要求图中不能包含权值总和为负值的回路。

Bellman-Ford算法思想

- 如图所示，回路(0, 1, 0)的长度为 -1.
- 这样 0 到 2 的最短路可以达到-∞.
- 如果存在这样的回路，就不能采用 Bellman-Ford 算法。
- 注意：带负边的回路只要权值总和非负，就不影响。



Bellman-Ford算法思想

- 假设现在有向图中有 n 个顶点，且不存在负权值回路，从顶点 v1 到 v2 如果存在最短路，则此路径最多只有 n - 1 条边。
- 如果超过 n - 1 条边，必定重复经过了同一个点，形成了一个回路。
- 由于回路权值之和非负，去掉这个回路，最短路径可以更优。

Bellman-Ford算法思想

- Bellman-Ford 算法可以构造出一个最短路径长度数组序列：
  - $dist^1[u]$，$dist^2[u], \cdots, dist^{n-1}[u]$
- 其中，$dist^1[u]$为从源点 v_0 到终点 u 只经过一条边的最短路径长度，并有$dist^1[u]$ = $edge[v_0][u]$
- $dist^2[u]$为从源点 v_0 出发最多经过两条边到达 u 的最短路径长度
- $dist^3[u]$为从源点 v_0 出发最多经过 3 条边到达 u 的最短路径长度
- 以此类推……
- 算法的最终目的是计算出$dist^{n-1}[u]$，为源点 v_0 到顶点 u 的最短路径长度。

Bellman-Ford算法思想

- 采用递推的方式计算 $dist^k[u]$
- 假设当前已经知道了 $dist^{k-1}[u]$
- 枚举 j，$dist^k[u] = \min\{dist^{k-1}[u],\ dist^{k-1}[j] + edge[j][u]\}$
- 这样就可以求出 $dist^{k-1}[u]$

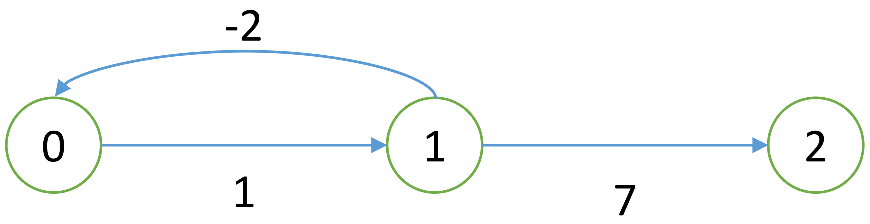
- 初始状态 $dist^1[u] = edge[v_0][u]$.
- 这样就可以递推了。

Bellman-Ford算法时间复杂度

- 总共要递推 n - 1 轮，每一轮中都要枚举每个点一次，每条边一次。
- 如果使用邻接矩阵存图，则时间复杂度为 O(n^3)
- 如果使用邻接表存图，则时间复杂度为O(n(n + m)).

Bellman-Ford算法判断负环

- 在求出 $dist^{n-1}$[] 之后，再对每一条边<u, v>判断一下：
  - dist[u] + edge[u][v] < dist[v]
- 如果成立，说明存在源点可达的负权值回路。



Bellman-Ford算法与Dijkstra算法比较

- 区别：
  - Dijkstra算法在求解过程中，源点到集合 S 内各顶点的最短路径一旦求出，之后就不会变了。之后修改的都是 T 集合中各顶点的最短路径长度。
  - Bellman-Ford算法在求解过程中，每次循环每个顶点的 dist[] 值都有可能要更改，也就是说源点到各顶点的最短路径长度一直要到算法结束才能确定下来。

下节课再见