

SPFA算法



主讲人：邓哲也



大纲

➤ Bellman-Ford算法的局限性

➤ SPFA算法

➤ SPFA算法判断负环

Bellman-Ford算法的局限性

- Bellman-Ford算法的时间复杂度较高，为 $O(n^3)$ 或 $O(nm)$
- 原因在于 Bellman-Ford 算法要递推 n 次。每次递推，扫描所有的边，在递推 n 次的过程中很多判断是多余的。

SPFA算法

- SPFA (Shortest Path Faster Algorithm) 算法是 Bellman-Ford 算法的一种队列实现，减少了不必要的冗余判断。
- SPFA算法的大致流程是用一个队列来进行维护。
 - 初始时，将源点加入队列。
 - 每次从队列中取出一个顶点，并对所有与它相邻的顶点进行松弛，若某个相邻的顶点松弛成功，则将其入队。
 - 重复这个过程，直到队列为空。

SPFA算法

- SPFA 算法，简单的说就是队列优化的 Bellman-Ford算法，是根据“**每个顶点的最短距离不会更新次数太多**”的特点来进行优化的。
- SPFA 算法可以在 $O(km)$ 的时间复杂度内求出源点到其他所有顶点的最短路径，并且可以处理负权值边。
- k 为每个顶点入队列的平均次数。可以证明对于通常的情况， k 为 2 左右。

SPFA算法

- 与 Bellman-Ford 算法类似，SPFA算法也使用 $\text{dist}[i]$ 数组存储源点 v_0 到顶点 v_i 的最短路径长度。
- 初始时， $\text{dist}[v_0] = 0$ ，其余均为 ∞ ，并将 v_0 入队列。
- 接下来每次取出队列头顶点 v ，扫描 v 出发的每条边 $\langle v, u \rangle$ ，权值为 w 。如果 $\text{dist}[u] > \text{dist}[v] + w$ ，更新 $\text{dist}[u]$ ，若顶点 u 不在队列中，还要把 u 加入队列。
- 重复直到队列为空。

SPFA算法判断负环

- 在 SPFA 中，如果一个顶点入队列的次数超过 n ，则表示有向图中存在负权值回路。
- 原理是：如果存在负权值回路，那么从源点到某个顶点的最短路径可以无限缩短，某些顶点入队列将超过 n 次。
- 因此，只需在 SPFA 算法中统计每个顶点入队列的次数，在取出队头顶点的时候，都判断其入队次数是否已经超过 n 次。

下节课再见