数据结构与算法图论算法

陈宇琪

2020 年 5 月 31 日

摘要

主要内容：最短路、最小生成树、图上动态规划。

DDL：2020-06-21

目录

1 Bellmanford 算法

1.1 边松弛

假设一条边满足 $d[e.v] < d[e.u] + e.w$，则更新 $d[e.v]$。

1.2 队列优化算法描述

- 初始将起点放入队列，并进入循环。
- 每次选取队首顶点 u。
- 对所有从 u 出发的边 $u \to v$ 进行变松弛。
- 如果节点 v 被更新，且 v 入队次数小于 n，则将 v 放入队尾。
- 如果一个点加入队列 n 次，则说明图中存在环。

* 详细算法可以参考 CSDN。

2 填表题

表 1: 四种最短路算法对比

	算法实质（贪心/动规/其他）	适用范围（单点源/多点源）	边权范围（正边权/皆可）	能否判负环（是/否）	算法复杂度
Dijkstra					
SPFA					
Floyd					
*Bellmanford					

* 请自学 Bellmanford 算法。

3 简答题

1、给出一个不等式组：

$$\begin{cases} x_1 - x_2 \le 0 \\ x_3 - x_1 \le 5 \\ x_3 - x_2 \le 6 \\ x_2 - x_4 \ge 0 \end{cases}$$

（1）构造方程组对应的差分约束系统。
（2）根据图的性质判断方程组是否有解？（简要说明理由）
*（3）构造原方程组的一组解。（修改差分约束系统的图结构，加入一个虚拟节点）
2、给出一个不等式组：

$$\begin{cases} x_3 - x_2 \le 0 \\ x_2 - x_1 \le 5 \\ x_3 - x_1 \ge 6 \end{cases}$$

（1）构造方程组对应的差分约束系统。

（2）根据图的性质判断方程组是否有解？（简要说明理由）

3、仔细阅读 1.2 中算法描述，并回答问题：

（1）什么是简单路径？

（2）为什么一个点只需要加入队列 n 次？（提示：最短路是简单路径）

（3）简述 Bellmanford 判负环的原理。

*（4）根据 1.2 描述的 Bellmanford 算法是否能够找出负环上的所有点？如果不能，请修改 Bellmanford 算法。（如果有多个负环，请找出所有最短路径长度为负无穷的点）

4、描述 Floyd 算法的状态设计和状态转移方程的具体含义。

4 编程题

1、（完整代码 + 算法描述）使用 Dijkstra 算法计算从 s 到 t 的最短路和次短路。

提示：请使用堆优化的 Dijkstra 算法，算法描述可以参考 PPT。

2、（完整代码 + 算法描述）假设给定一个有向无环图，求宋 s 出发到 t 的简单路径条数。

路径计数：假设对于两条从 s 到 t 的简单路径，只要经过的边有一条不一样，则两条路径认为是不同的。

提示 1：你可以认为图中不存在重边，如果你能够处理重边当然更好。

提示 2：对于有向无环图而言，常见做法是先求拓扑排序。

提示 3：一个简单版本，你可以认为所有边 (u, v) 满足 $u < v$。