A星、Floyd、Bellman-Ford与SPFA

前置知识:
讲解059-建图、链式前向星
讲解062-宽度优先遍历
讲解064-Dijkstra算法
讲解059~讲解065都是【必备】课程有关图的内容，建议从头开始学习

注意:
【必备】标签下的课程，都是最基础、最高频的内容
有关图的更多内容，会在后续【扩展】、【挺难】标签下的课程中继续讲述

A星、Floyd、Bellman-Ford与SPFA

A*算法，指定源点，指定目标点，求源点到达目标点的最短距离
增加了当前点到终点的预估函数
在堆中根据 从源点出发到达当前点的距离+当前点到终点的预估距离 来进行排序
剩下的所有细节和Dijskra算法完全一致

预估函数要求：当前点到终点的预估距离 <= 当前点到终点的真实最短距离
预估函数是一种吸引力
1）合适的吸引力可以提升算法的速度，吸引力过强会出现错误
2）保证 预估距离 <= 真实最短距离 的情况下，尽量接近真实最短距离，可以做到功能正确 且 最快

预估终点距离经常选择：
曼哈顿距离
欧式距离
对角线距离

A星、Floyd、Bellman-Ford与SPFA

Floyd算法，得到图中任意两点之间的最短距离
时间复杂度O(n^3)，空间复杂度O(n^2)，常数时间小，容易实现
适用于任何图，不管有向无向、不管边权正负、但是不能有负环（保证最短路存在）

过程简述:
distance[i][j]表示i和j之间的最短距离
distance[i][j] = min (distance[i][j] , distance[i][k] + distance[k][j])
枚举所有的k即可，实现时一定要最先枚举跳板！

A星、Floyd、Bellman-Ford与SPFA

Bellman-Ford算法，解决可以有负权边但是不能有负环（保证最短路存在）的图，单源最短路算法

松弛操作
假设源点为A，从A到任意点F的最短距离为distance[F]
假设从点P出发某条边，去往点S，边权为W
如果发现，distance[P] + W < distance[S]，也就是通过该边可以让distance[S]变小
那么就说，P出发的这条边对点S进行了松弛操作

Bellman-Ford过程
1，每一轮考察每条边，每条边都尝试进行松弛操作，那么若干点的distance会变小
2，当某一轮发现不再有松弛操作出现时，算法停止

A星、Floyd、Bellman-Ford与SPFA

Bellman-Ford算法时间复杂度
假设点的数量为N，边的数量为M，每一轮时间复杂度O(M)
最短路存在的情况下，因为1次松弛操作会使1个点的最短路的边数+1
而从源点出发到任何点的最短路最多走过全部的n个点，所以松弛的轮数必然 <= n - 1
所以Bellman-Ford算法时间复杂度O(M*N)

重要推广：判断从某个点出发能不能到达负环
上面已经说了，如果从A出发存在最短路（没有负环），那么松弛的轮数必然 <= n - 1
而如果从A点出发到达一个负环，那么松弛操作显然会无休止地进行下去
所以，如果发现从A点出发，在第n轮时松弛操作依然存在，说明从A点出发能够到达一个负环

A星、Floyd、Bellman-Ford与SPFA

Bellman-Ford + SPFA优化（Shortest Path Faster Algorithm）
很轻易就能发现，每一轮考察所有的边看看能否做松弛操作是不必要的
因为只有上一次被某条边松弛过的节点，所连接的边，才有可能引起下一次的松弛操作
所以用队列来维护 “这一轮哪些节点的distance变小了”
下一轮只需要对这些点的所有边，考察有没有松弛操作即可

SPFA只优化了常数时间，在大多数情况下跑得很快，但时间复杂度为O(n*m)
看复杂度就知道只适用于小图，根据数据量谨慎使用，在没有负权边时要使用Dijkstra算法

网上说，SPFA已死。有时候死了，有时候诈尸了，称为薛定谔的SPFA，这是啥意思？

A星、Floyd、Bellman-Ford与SPFA

Bellman-Ford + SPFA优化的用途
1）适用于小图
2）解决有负边（没有负环）的图的单源最短路径问题
3）可以判断从某个点出发是否能遇到负环，如果想判断整张有向图有没有负环，需要设置虚拟源点
4）并行计算时会有很大优势，因为每一轮多点判断松弛操作是相互独立的，可以交给多线程处理

注意：
SPFA的另一个重要的用途是解决“费用流”问题，当然也可以被Primal-Dual原始对偶算法替代
这一内容会在【挺难】标签下的课程里讲述
“费用流”问题在大厂笔试、面试中是冷门内容，但是致力于比赛的同学是必须要掌握的

A星、 Floyd、 Bellman-Ford与SPFA

题目1
A*算法模版
A*算法 vs Dijkstra算法
采用对数器验证

左程云

A星、 Floyd、 Bellman-Ford与SPFA

题目2
Floyd算法模版（洛谷）
测试链接 : https://www.luogu.com.cn/problem/P2910

左程云

A星、Floyd、Bellman-Ford与SPFA

题目3
Bellman-Ford算法应用（Leetcode）
测试链接 : https://leetcode.cn/problems/cheapest-flights-within-k-stops/

A星、Floyd、Bellman-Ford与SPFA

题目4
Bellman-Ford + SPFA优化（洛谷）
给定一个 n个点的有向图
请求出图中是否存在从顶点 1 出发能到达的负环
负环的定义是：一条边权之和为负数的回路。
测试链接 : https://www.luogu.com.cn/problem/P3385