



简单图论算法讲解

GAREN 2019.04

最小生成树

- 在 n 个点的无向图中删至只有 $n - 1$ 条边，并整图连通，这个新图就是生成树。最小生成树是边权最小的生成树。
- 最小生成树算法有两种：prim和kruskal。
- Prim算法是蓝白点思想，每次贪心取已开辟点集中距离最小的点来加入最小生成树，同时并入已开辟点。适用于稠密图。有堆优化的prim。
- Kruskal算法把所有的边按边权从小到大排序，然后按顺序添边，能添就添，不能添就跳至下一个。利用并查集可以简单的实现。适用于稀疏图。
- 我们重点掌握kruskal算法，prim思路懂了就好。

最短路算法

- 最短路算是OI里面最基础的图论了。可用于求解点权或边权的最短路。
- 这里介绍两种最短路算法：SPFA和dijkstra堆优化。

SPFA

- 关于SPFA（它死了）
- SPFA说白了就是用队列储存的bellman-ford算法。最坏复杂度 $O(nm)$ ，随机图快。
- 虽然SPFA死了，但是仍然可以借助它的思想求最长路、判负环、判正环。
- 思路是用一个队列去存需要松弛的节点，依次拿出未松弛的节点进行松弛更新最短路。
- 唯一注意的是进队列的元素需要判重。
- SPFA可以跑负权图。但是是正权图的时候最好别用SPFA！

SPFA衍生算法

- 判负环：用SPFA跑最短路中，若一个点进队超过 n 次，则存在负环。（无最短路）
- 最长路：把dist数组初始赋为极小值，改下松弛条件，就可以跑最长路了。
- 判正环：用SPFA跑最长路的时候，同样当一个点进队超过 n 次就有正环。

Dijkstra算法

- Dijkstra算法是单源最短路算法，比SPFA稳定多了。
- Dijkstra算法思想是取出离当前已开辟节点中最近的节点来松弛，然后把它归为已开辟节点，这样循环走 $n - 1$ 次。
- 普通dijkstra需要遍历一遍所有点取得最小dist，而堆优化dijkstra就把所有dist和对应下标存到堆里面去，维护一个以dist为关键字的小根堆。
- 普通dijkstra是 $O(n^2)$ 的，堆优化后是 $O(m + n \log n)$ 。
- 注意：dijkstra算法不适用于负权图！负权图只能用SPFA。
- 对应题目：很多很多

最短路计数

- 最短路计数只是在最短路的基础上维护了计数的信息，定义一个cnt数组来dp。
- 当一个点被松弛，到达该点的最短路数就继承被松弛的点。
- 当松弛的路跟最短路一样长，到该点的最短路数就加上被松弛点的最短路数。
- 但不是对所有节点计数的时候，好像其他的要清零。这点我不清楚。
- 对应题目：P1608和P1144。
- NOIP2017 D1T3 30pts部分分

Floyd算法

- Floyd算法是最简单的多源最短路算法，复杂度是稳定的 $O(n^3)$ 。
- 代码唯一要注意的是先枚 k 再枚 i 和 j 。因为floyd是三维dp滚动后的算法。
- Floyd的本质需要灵活运用，因为实在是太灵活了。对应题目：P1119、P1730、T21904
- 顺便介绍一个黑科技：
- bitset优化floyd解决传递闭包问题，复杂度 $O(\frac{n^3}{32})$ 。对应题目：P2881、P4306

拓扑排序


- 能拓扑排序的叫有向无环图 (DAG)，反之亦然。
- 拓扑排序一般用bfs跑，记录所有点的入度，先把入度为0的入队，然后与之相连的点入度-1，如果减完入度为0则入队。跑出来的顺序就是拓扑序。
- 一般用于有先决性的图论问题。听说还能判环。
- 对应题目：P1983、P1038

Tarjan算法

- Tarjan算法能用来判断强连通分量。强连通分量可理解为一个环或不成环的一个点。
- Tarjan用dfn数组维护dfs序，用low数组维护往上翻能到达的最小dfn，并且用栈储存点，vis数组判断一个点是否在栈中。
- 如何更新low数组？
- 当访问到没访问的节点，先递归从它开始跑，然后用它的low更新。
- 当访问到栈中的节点，直接拿它的dfn更新。
- 如果一个点的dfn等于low，那么从栈顶到它的所有节点都属于一个强连通分量。
- 不会证明

Tarjan算法应用

- Tarjan算法可以把一个强连通分量缩成一个点，以此建立新图。新图是DAG。
- 也可以求一个图的最大环。各个强连通分量的size取最大值即可。
- 根据新图的入度和出度，有时能做一些题。
- 对应题目：P2341、P3387、P2746
- 与tarjan算法对应的有很多拓展内容：割点、桥、点双、边双等。（一般不用会）



再见