图论入门

hzq84621

- 最近几年图论考的还是不少的。
- 不过思维难度高的题并没有。
- 也是因为光图论考不出什么东西，一般作为DP的附庸出现。

最短路

- 三种
- floyd,dijkstra,spfa
- 最常用的还是堆优化dijkstra，如果不是骗骗不承认SPFA的外国人，一般能用dij就用dij。
- spfa可以用网格图卡掉。
- 另外还有个贝尔曼福德，那个东西除了判负环没什么吊用。
- 另外dij不适用于存在负权值的图（可能存在黑科技可以对付特殊的存在负权值的图）

差分约束

- 存在一类问题，给出若干个诸如a[i]-a[j]<=w的限制，和一个a[S]=x的初始条件，求a[T]的最大值。
- 假如从DP角度考虑的话，不难发现DP方程dp[j]=max(dp[i]+w)的形式和最短路的过程十分相似。
- 因此所有最短路的算法都可以直接套在这个DP的过程上，即为差分约束。
- 与常规线性规划区别如下：
- 1、不等号必定单向，不会同时出现上下界
- 2、每个限制只存在2个元素

成环DP&最短路

- 假如将每个DP状态当做图上的一个点，将DP的转移作为点与点之间的边处理。
- 常规DP的图都是拓扑图，但是存在一类奇葩DP的状态转移会成环
- 对于基于贪心的DP来说，往往都可以用最短路的做法跑，当然如果DP状态形成的图是个拓扑图的话不需要这么麻烦。
- 对于基于计数的DP来说，一般边权都不会大于1，否则就会出现结果无限大，所以能用图论解决的都是那种概率DP
- 这种时候就需要算出每个状态转移到自己的贡献，并且去掉这部分贡献之后再转移到其他状态。

Shoot-out

- 有N个牛仔进行决斗，N个人围成一圈依次朝自己选定的人开枪。
- 每个人存在射击命中率p[i]
- 可以选择不开枪。
- 每个人绝对聪明。假如存在多种让自己存活概率最大的方案会随机选择一种。
- 这个过程重复到只剩一个人存活。
- 求每个人存活的概率
- n<=13

拆点&缩点

- 拆点就是一个点需要有k种不同状态时，把1个点拆成k个点处理。
- 从DP的角度考虑就是dp[i][j]假如只考虑i不好维护，那么就把第i个点拆成j个每个点表示一个状态。
- 缩点就是存在一些例如每个点只能经过一次的条件。
- 这种时候拓扑图很好处理，但是环出于这些限制不好处理，所以需要把每个环缩成一个点。

另外一个常见的处理

- 有的时候分层图会有这样的要求。
- 上一层的第[l,r]个点向下一层的第[L,R]个点连边。
- 这个时候可以在中间设一个点，[l,r]向它连边,它向[L,R]连边
- 边数量就从n^2减少到了n
- 注意前面说到DP和图的联系，所以这个做法也适用于DP转移

Tarjan

- 有的时候我们需要判断有向图中是否存在环，可以用Tarjan解决。
- Tarjan基于DFS来找出环，从而进行缩点。
- NOIP来讲的话会背板子就行了。

割边&割点

- 割点的定义是去掉这个点后这个图不再联通
- 割边的定义是去掉这条边后这个图不再联通
- 若dfn[v] > low[u]则这条边为割边
- 若low[v] >= dfn[u]则这个点为割点，注意开始DFS的那个
- 点需要有两个这样的v才能作为割点

bridge

- 给定一个图。
- 在这个图上加一条边，求最小割边数
- $n<=100W,m<=100W$

图&树

- 有些针对图上路径的多组询问问题，一般都是不能直接处理的。
- 这种情况答案往往会出在最小生成树上

货车运输

- 给定一个图。
- Q次询问，每次询问从x[i]走到y[i]的路径中，边权最小值的可能最大值是多少。
- n<=10000,m<=50000,Q<=30000

故乡的梦

- 给出一个图，S和T
- 每次询问，去掉一条边时，从S到T的最短路是多少
- n,m<=200000,Q<=200000

GL&HF