For HuanTwoFat, iBlackSun, Yanaz

数据结构图论与网络流杂谈

基本的姿势

- 树状数组=线段树=平衡树=字典树

- 持久化线段树
- 延迟节点建立
- 主席树?
- 线段树和块状链表的最主要差别：统计量是否上传

并查集

- 按秩合并=>O(logN)->可以持久化
- 路径压缩=>O(alpha(N))->不能持久化

- 完全持久化->线段树代替数组->O(log^2N)
- 只有回退->用栈记录每一次的合并->均摊O(logN)

- 莫队？=>回退式莫队
- 并查集可以搞一些奇怪的矩阵题

离线动态图

- 方法一：
- 时间维度上CDQ分治
- 用回退并查集或者DFS缩点
- 缩点的话，注意要删掉不必要的点
- 参考：CF-gym-100551A和北大校赛C

离线动态图

- 方法二：最晚删除生成森林
- 用LCT，加边时破环
- 删边时候直接删就好了，因为删除时间是任意环上最晚的，所以破开的两个分量不可能连通
- 通常的写法会化边为点，但是可以不这么写（很烦，看码力）
- 参考BZOJ4025，还有算法教室的生成树(POJ3522加强版)

DFA化简

- 先按基本的终结、非终结划分
- 每轮找到和其他人不同的，划分成新的一类
- 注意每一轮找不同的时候不能访问这一轮新分出去的类
- 分到不能分为止
- 参考CF-gym-100553E

分组：合并不同的算法

- 典型的是北大校赛的J
- 大组组数少元素多，用双指针
- 小组组数多元素少，用十字链表
- 大小组之间，枚举小组元素用二分在大组中找

- 还有15年NOI的最后一题
- 小质数状压，大质数DP

KDTREE

- 创建：每次找方差最大的维度，用nth_element劈分
- 插入：类似替罪羊，找到第一个不平衡点暴力重建。也可以直接重建整个子树。
- 删除：要找同维度的下一个后继，很难写，建议改空权值，不要真删除。
- 查询：判断矩形的交。
- KDTree一般用来水题的，正解都是其他算法。

树上姿势

- 典型如CF375D
- 先递归处理非重孩子，返回后清空数据
- 然后递归处理重孩子，返回后不清理数据，而是把根节点和非重孩子的数据加到上面去
- 最后解决在根的查询
- 复杂度O(NlogN)
- 不需要树上莫队……

归并与划分树

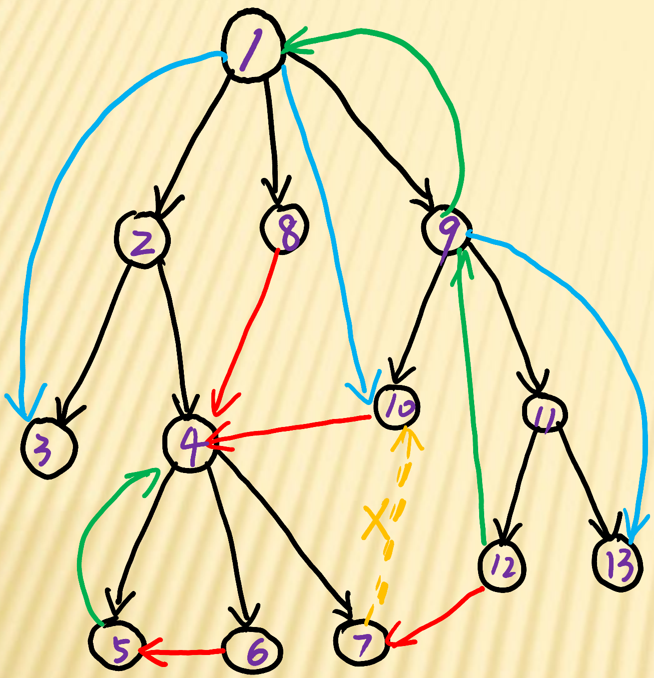
- 典型如CF453E
- 利用清零标记
- 被清零的一段，按从零到满需要的时间排序，记录前缀和，可以直接出答案
- 清零时间不一致时直接暴力即可，均摊时间还是O(NlogN)

利用数学简化数据结构

- 典型题目是算法教室⑨的extra
- 记录选取若干张牌使得余数为x的方法数
- 很容易想到CDQ分治……
- 然而加牌的操作是可逆的，所以不用CDQ

- 也可以DP处理每种卡片有N张时候对方法数的影响，打一个大表

TARJAN算法的四类边



→ 树枝边

→ 横叉边

→ 前向边

→ 后向边

12 时间

-X→ 不可能边

其中，前向边和横叉边只有有向图有，无向图没有。

一些几乎不可能考的算法

- Dominator Tree
- Stoer Wagner全局最小割
- 带修改树上莫队
- Steiner Tree
- 最小树形图的朱刘算法
- 考出来纯坑人的，有板子就A，没板子就跪。

数据结构的思考方法

- 首先应该看在线离线，只要能离线先考虑是不是标准的莫队、CDQ分治、离线DFS，能不能块状链表水过。
- 然后考虑暴力算法应该怎么写，能不能直接加个位图优化就水过去（特别是图论题），是不是可以分组根号。
- 最后等数据的性质了解清楚了再枚举能用的数据结构代入。

后缀自动机拓扑排序

- 按照顶点的max值进行计数排序

线性规划的对偶

$$\min \mathbf{c}^{T}\mathbf{x}$$
$$\text{st.}\begin{cases}\mathbf{Ax} \ge \mathbf{b} \\ \mathbf{x} \ge 0\end{cases}$$

$$\max \mathbf{b}^{T}\mathbf{y}$$
$$\text{st.}\begin{cases}\mathbf{A}^{T}\mathbf{y} \le \mathbf{c} \\ \mathbf{y} \ge 0\end{cases}$$

强对偶定理

- 对偶问题的任一可行解，其目标函数值小于原始问题的任一可行解对应的目标函数值。
- 两个问题或者都有有限的最优解，或者都没有。
- 如果两个问题存在最优解，在此最优解下两目标函数必相等。

互补松弛型定理

- 对于最优解X, Y（向量）和任意下标j
- 如果X[j]非0，那么Y的第j个约束一定为紧约束。
- 反之，如果Y的第j个约束不紧，那么X[j]=0

$$\forall 1 \le j \le n, \left(\sum_{i=1}^{m} A_{ij} Y_i - c_j \right) X_j = 0$$

全幺模矩阵

- 定理：有向图和无向二分图的关联矩阵为全幺模矩阵。
- 定理：如果线性规划的矩阵A是全幺模矩阵，则此线性规划之解必为整点解。
- 如：最短路，二分图匹配，网络流。

常见问题的规划方程

- 最短路：

$$\min \sum_{(i,j)\in A} w_{ij} x_{ij}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} 1, & i = s, \\ -1, & i = t, \\ 0, & i \neq s,t. \end{cases}$$

$$x_{ij} \geq 0.$$

常见问题的规划方程

- 最短路的对偶：

$$\max(u_t - u_s)$$

$$\text{s.t. } u_j - u_i \leqslant w_{ij}, \forall (i,j) \in A.$$

$$x_{ij}(u_j - u_i - w_{ij}) = 0, \forall (i,j) \in A.$$

- 标号u含义为（相对s）距离

常见问题的规划方程

- 最大流：

$$\max v$$

$$\text{s.t.} \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} v, & i = s \\ -v, & i = t, \\ 0, & i \neq s, t, \end{cases}$$

$$0 \leqslant x_{ij} \leqslant u_{ij}, \qquad \forall (i, j) \in A.$$

- 最大流中流量是变量，一般不用对偶算法

常见问题的规划方程

- 费用流：

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} v, & i = s, \\ -v, & i = t, \\ 0, & i \in V, i \neq s,t, \end{cases}$$

$$0 \leqslant x_{ij} \leqslant u_{ij}, \qquad (i,j) \in A.$$

- 这里流量是常数，其对偶

常见问题的规划方程

$$\max w(\pi, z) = \sum_{i \in V} d_i \pi_i - \sum_{(i,j) \in A} u_{ij} z_{ij}$$

$$\text{s.t. } \pi_i - \pi_j - z_{ij} \leqslant c_{ij}, \qquad (i,j) \in A,$$

$$z_{ij} \geqslant 0, \qquad (i,j) \in A.$$

- 式中π是节点流量守恒式的对偶变量，z是流量上界的对偶变量。

常见问题的规划方程

- 由互补松弛性定理:

$$x_{ij}(\pi_i - \pi_j - z_{ij} - c_{ij}) = 0, \qquad (i,j) \in A,$$
$$z_{ij}(x_{ij} - u_{ij}) = 0, \qquad (i,j) \in A.$$

- 或写成:

当 $\pi_i - \pi_j < c_{ij}$ 时， $x_{ij} = 0$；
当 $\pi_i - \pi_j > c_{ij}$ 时， $x_{ij} = u_{ij}$；
当 $0 < x_{ij} < u_{ij}$ 时， $\pi_i - \pi_j = c_{ij}$.

$$z_{ij} = \max\{\pi_i - \pi_j - c_{ij}, 0\}, \qquad (i,j) \in A,$$

常见问题的规划方程

- 注意最短路问题本身也是费用流。所以π可以当成距离看待。
- 定义每条边的既约代价： $c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j$
- 费用流问题的最优解一定存在一棵既约代价为0的生成树。

常见问题的规划方程

- 不知道怎么构图的时候，一定先列出规划方程，然后取对偶，写出标号变量和互补松弛型式子。
- 它们不一定有意义，但是可以帮助构图。
- 最小割的构图什么的，我们队反正残了，没的教……

变分法

- 变分法和PDE在物理上挺常用的，计算机上嘛……
- 遇到题直接欧拉方程打脸就行

$$x_i = x_i(t) \text{ unknown} \qquad \dot{x}_i = \frac{dx_i}{dt}$$

$$\text{opt} \int_{t_0}^{t_1} F(x_i, \dot{x}_i, t)\,dt$$

$$\text{sol: } \forall i, \frac{d}{dt}\frac{\partial F}{\partial \dot{x}_i} - \frac{\partial F}{\partial x_i} = 0$$

LCS

- 如果已经匹配的位为0，匹配x的最低位
- 如果已经匹配的位不为0，用每一段最低的1换下同段末的1
- 如AXBCDA,BDA,此时结果是001010=>100011
- 于是计算的结果就是在保证最长长度的情况下可能有的最低位数分布
- 注意：如果要求出LCS的值而非长度，此时会出现错误

LCS

```
//位图算LCS(1个数对，位置不对)
//match[ch][i]是ch的出现位图
void BitsetLCS(ul A[], char ch){
  int bottom = 1, top; ul x, y;
  for(int i = 0; i < nword; i ++) {
    y = A[i]; x = y | match[ch][i];
    top = (y >> (32 - 1)) & 1;
    y = (y << 1) | bottom;
    if(x < y)  top = 1;
    A[i] = x & ((x - y) ^ x);
    bottom = top; }}
```

- 注意如果DP套DP的话，这里转移矩阵是三角矩阵。