图论相关知识点

罗雨屏

清华大学　交叉信息研究院

2014 年 10 月 2 日

基础概念

- 路径
- 简单路径（点、边）
- 生成树
- 连通分量
- 拓扑图（有向无环图，DAG）

基础知识

- 分类
  - 给定起点，给定终点
  - 给定起点，所有终点
  - 所有点对
- 算法
  - Dijkstra $O(n^2)$
    - 堆优化 $O((n+m)\log n)$
  - SPFA $O(km)$
    - 队列长度？循环队列
  - Floyd $O(n^3)$
    - 只适合求所有点对（或者点数特别少）
    - 注意三重循环的顺序
  - 拓扑图：DP $O(n+m)$
  - 思考：如何保存路径

基础知识

- 分类
  - 给定起点，给定终点
  - 给定起点，所有终点
  - 所有点对
- 算法
  - Dijkstra $O(n^2)$
    - 堆优化 $O((n+m)\log n)$
  - SPFA $O(km)$
    - 队列长度？循环队列
  - Floyd $O(n^3)$
    - 只适合求所有点对（或者点数特别少）
    - 注意三重循环的顺序
  - 拓扑图：DP $O(n+m)$
  - 思考：如何保存路径
    - 记录前驱

Dijkstra

- 只适用于正权
- 若 x 为所有未确定最短路的点中，“可能”距离 s 最近的点
  - 可能：只经过一条边即到达已确定最短路的点
- 则可以确定到 x 的最短路
  - 因为其只会经过一条边到达一个已经确定最短路的点
- 维护到每个点的“可能”最近的距离
- 复杂度 $O(n^2)$
  - 堆优化 $O(n + m\log m)$

Bellman-Ford

- 枚举每条边，进行松弛操作
- 共有 n 轮，$O(nm)$

Algorithm 1 Bellman-Ford Algorithm

```
for i = 1 → n do
    for each edge u → v do
        if d_u + w_{u,v} < d_v then
            d_v ← d_u + w_{u,v}
        end if
    end for
end for
```

SPFA

- 观察：如果 d_u 在第 k 轮中没有改变，那么 $u \to v$ 这条边在第 $k+1$ 轮不可能更新 d_u
- 利用：队列
  - 用一个队列维护当前所有可能可以更新 v 的 u
  - 当一个点被更新之后加入队列中
  - 队列长度：循环队列
- 负圈环的判断方法：如果一个点入队超过 n 次
- 栈式 SPFA 与队列式 SPFA
  - 栈：适合权值负数较多的情况
  - 结合起来：如果更新后的距离比队首小，则放在队首，否则放入队尾（Small Label First）

SPFA cont'd

- 加入 SLF 后判断负圈环会有问题
  - 最短路经过的边的数目必定小于 n
  - 记录最短路经过的边的数目即可
- SLF 的精髓：让权值小的点尽量放在队列前面
  - 用一个堆来维护？
  - 如果用的不是权值“一定最小”的，而是“相对较小”的点？
  - 每次判断：如果队首值大于平均值，则将队首放在队尾（Large Label Last）

Floyd

Algorithm 2 Floyd Algorithm

```
for k = 1 → n do
    for i = 1 → n do
        for j = 1 → n do
            f[i,j] ← min(f[i,k] + f[k,j], f[i,j])
        end for
    end for
end for
```

- 理解：DP
  - 中途不经过标号 $\geq k$ 的点时，$i \to j$ 的最短路长度

比较

- SPFA、Dijkstra、Floyd 的优劣势在哪里？
- 如何卡 SPFA ？

比较

- SPFA、Dijkstra、Floyd 的优劣势在哪里？
- 如何卡 SPFA ？
  - 网格图
  - 拓扑图

比较

- SPFA、Dijkstra、Floyd 的优劣势在哪里？
- 如何卡 SPFA ？
  - 网格图
  - 拓扑图
- 如何使 Dijkstra + heap 的复杂度达到上界？

拓扑图路径计数 Problem

- 热身题
- 给定一个拓扑图 $G=(V,E)$ 以及图中两点 S,T ，求 S 到 T 有多少条不同路径
-
- $|V| \leq 10^5, |E| \leq 3 \times 10^5$

拓扑图路径计数 Solution

- 还是简单的 dp
- f_i 表示 S 到点 i 的路径数目
  - $f_i = \sum_{j \to i} f_j$
- 答案即为 f_T

两点间路径计数 Problem

- 给定一个有向图 $G = (V, E)$ 以及图中两点 S, T ，求 S 到 T 有多少条长度恰好为 k 的路径
-
- $|V| \le 10^2, |E| \le 10^3$

两点间路径计数 Solution

- 矩阵乘法 $(AB)_{i,j} = \sum_{k} A_{i,k}B_{k,j}$
- 令 M 为 G 的邻接矩阵，则 $(M^k)_{i,j}$ 即为从 i 走恰好 k 步到 j 的方案数

两点间路径计数 Solution

- 矩阵乘法 $(AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$
- 令 M 为 G 的邻接矩阵，则 $(M^k)_{i,j}$ 即为从 i 走恰好 k 步到 j 的方案数
- 类似于快速幂的矩阵乘法 $O(n^3 \log k)$

两点间最短路计数 Problem

- 给定一个有向图 $G=(V,E)$ ，以及图中两点 S,T ，求从 S 到 T 的最短路条数
-
- $|V| \leq 10^3, |E| \leq 3 \times 10^5$

两点间最短路计数 Solution

Definition (最短路网)

- 所有满足 $dist_{S,x} + w_{x,y} + dist_{y,T} = dist_{S,T}$ 的边 (x, y) 组成的图

- 所有 S 到 T 的最短路上的边一定在最短路网上
- 所有最短路网上 S 到 T 的路径一定是 S 到 T 的最短路
- 统计最短路网上 S 到 T 的路径数目即可

两点间最短路计数 Solution

Definition (最短路网)

- 所有满足 $dist_{S,x} + w_{x,y} + dist_{y,T} = dist_{S,T}$ 的边 (x, y) 组成的图

- 所有 S 到 T 的最短路上的边一定在最短路网上
- 所有最短路网上 S 到 T 的路径一定是 S 到 T 的最短路
- 统计最短路网上 S 到 T 的路径数目即可
  - 拓扑图

SDOI 集训 Problem

- 给定一个图 $G = (V, E)$
- 给定 k 个点 $v_1, v_2, \ldots, v_k$
- 求从 1 到 n 经过不少于 L 条边不超过 R 条边、给定的 k 个点都经过的路径数目
-
- $|V| \leq 50, k \leq 4$

SDOI 集训 Solution

- 两种限制
  1. 路径长度的限制
  2. 必经点的限制

SDOI 集训 Solution

- 两种限制
  1. 路径长度的限制
  2. 必经点的限制
- 关于路径长度的限制
  - 只要求出 $\sum_{i=0}^{R} M^i$ 即可，其中 M 表示邻接矩阵
  - 构造矩阵

$$\begin{bmatrix} M & I \\ 0 & M \end{bmatrix}$$

    即可

SDOI 集训 Solution

- 两种限制
  1. 路径长度的限制
  2. 必经点的限制
- 关于路径长度的限制
  - 只要求出 $\sum_{i=0}^{R} M^i$ 即可，其中 M 表示邻接矩阵
  - 构造矩阵

$$\begin{bmatrix} M & I \\ 0 & M \end{bmatrix}$$

    即可
- 关于必经点的限制
  - P.I.E
  - 对于任意一个 $\{v_1, v_2, \ldots, v_k\}$ 的真子集，求出不能经过这个子集的路径数目

差分约束系统 Problem

- 有 n 个变量 $x_1, x_2, \ldots, x_n$ ， m 个限制
- 每个限制形如 $x_i - x_j \leq k$
- 求一组可行的方案, 或者输出无解
- $n \leq 10^3, m \leq 10^5$

差分约束系统 Solution

- 变形

$$x_i - x_j \le k \leftrightarrow x_i \le x_j + k$$

差分约束系统 Solution

- 变形

$$x_i - x_j \le k \leftrightarrow x_i \le x_j + k$$

- 对于一个图, 令 d_i 表示点 0 到 i 的最短路长度，有：

$$d_i \le d_j + w_{j,i}$$

差分约束系统 Solution

- 变形

$$x_i - x_j \leq k \leftrightarrow x_i \leq x_j + k$$

- 对于一个图，令 d_i 表示点 0 到 i 的最短路长度，有：

$$d_i \leq d_j + w_{j,i}$$

- 把 x_i 看成是 d_i ，转为求最短路
  - 若限制 $x_i - x_j \leq k$ ，则添加一条 j 到 i 的边，权值为 k
  - 初始时点 0 向任意点连一条权值为 0 的边
  - 无解 $\leftrightarrow$ 负权环

最小乘积路径

- 给定一个图 $G = (V, E)$ 以及图中的两个点 S, T
  - 边正权
- 求所有 S 到 T 的路径中，路径中的边权乘积最小的路径
-
- $|V| \leq 10^5, |E| \leq 3 \times 10^5$

最小乘积路径

- 给定一个图 $G = (V, E)$ 以及图中的两个点 S, T
  - 边正权
- 求所有 S 到 T 的路径中，路径中的边权乘积最小的路径
-
- $|V| \le 10^5, |E| \le 3 \times 10^5$
-
- 技巧：用 $\log$ 把乘除化为加减

k 短 Walk Problem

Definition (Walk and Path)

- Walk 允许经过一个点两次
- Path 只允许经过一个点一次

- 给定一个边正权的无向图 $G = (V, E)$
- 求一条 S 到 T 的 Walk
- 使得这条路径在所有可能路径中的排名为 k
-
- $n, k \leq 100, m \leq 1000$

k 短 Walk Solution

- 考虑 Dijkstra （with heap）
- 为了求第 k 短 Walk，对于每个点应该保存到这个点的前 k 小 Walk 的距离
  - 也就是堆中允许存在多个点表示一个图中的顶点
  - 最多堆中有多少个元素：$O(nk)$
- 复杂度 $O((nk+m)\log nk)$

k 短 Path Problem

- 几乎与上题相同，除了这次求的是 Path
-
- $n, k \leq 50, m \leq 200$

k 短 Path Solution

- 二分答案
- 每次查询长度不超过 L 的 Path 的数目是否超过 k
- 同样是 NPC 问题，剪枝
- code

```
if curDist + dist(u, i) > L :
    return
```

k 短 Path Solution 2

- 利用 k 短 Walk 的结果
- 找一个较小的 k' 满足前 k' 短 Walk 中有至少 k 个 Path
- 如何找 k' ？

k 短 Path Solution 3

- 令 P_i 表示第 i 短的 Path
- 则 P_k 不能与 $P_1, P_2, \ldots, P_k$ 相同
  - 至少要有一条边不同
- 试图将 P_i 中的一条边删去
- 时间复杂度 $O(n^{k-1})$

变种 I Problem

- 给定一个正权图 $G = (V, E)$
- 给定 S, T ，你可以将一条边的权值减小一半（上取整），使得 S 到 T 的最短路最短
- 求最小距离
- $|V| \leq 10^5, |E| \leq 3 \times 10^5$

变种 I Solution

- 拆点
  - v 分为 v_1 和 v_2
  - 原图有 $e = s \to t$ 的边，连接 $s_1 \to t_1$ 以及 $s_2 \to t_2$，权值 w_e，连接 $s_1 \to t_2$ 权值 $\frac{1}{2}w_e$
- 求新图中 S_1 到 T_2 的最短路

变种 I Solution

- 拆点
  - v 分为 v_1 和 v_2
  - 原图有 $e = s \to t$ 的边，连接 $s_1 \to t_1$ 以及 $s_2 \to t_2$ ，权值 w_e ，连接 $s_1 \to t_2$ 权值 $\frac{1}{2}w_e$
- 求新图中 S_1 到 T_2 的最短路
- 为什么这样做是对的？
  - 整个图分为两层，一旦走到了第二层就不可能再走回第一层
  - 只有在从第一层走到第二层时才会走一条权值为一半的边

变种 I Solution

- 拆点
  - v 分为 v_1 和 v_2
  - 原图有 $e = s \to t$ 的边，连接 $s_1 \to t_1$ 以及 $s_2 \to t_2$，权值 w_e，连接 $s_1 \to t_2$ 权值 $\frac{1}{2}w_e$
- 求新图中 S_1 到 T_2 的最短路
- 为什么这样做是对的？
  - 整个图分为两层, 一旦走到了第二层就不可能再走回第一层
  - 只有在从第一层走到第二层时才会走一条权值为一半的边
- Any other solution?

必经点、必经边 Problem

- 给定一个图 $G = (V, E)$
- 求 S 到 T 的最短路上的必经点
-
- $|V| \le 10^5, |E| \le 3 \times 10^5$

必经点、必经边 Solution

- 求出最短路网 $\rightarrow$ 化除“最短路”这个条件
- 如何求必经点、必经边?
  - 对于每个点 V
  - 求出从 S 到 v 的路径条数 P_v
  - 求出从 v 到 T 的路径条数 Q_v
  - 令 S 到 T 路径条数为 Z
  - 对于一个点 v，v 是必经点等价于 $Z = P_v Q_v$
  - 对于一条边 $e : u \rightarrow v$，e 是必经边等价于 $Z = P_u Q_v$
- 路径条数太大, 要写高精?

必经点、必经边 Solution

- 求出最短路网 $\rightarrow$ 化除“最短路”这个条件
- 如何求必经点、必经边?
  - 对于每个点 V
  - 求出从 S 到 v 的路径条数 P_v
  - 求出从 v 到 T 的路径条数 Q_v
  - 令 S 到 T 路径条数为 Z
  - 对于一个点 v，v 是必经点等价于 $Z = P_vQ_v$
  - 对于一条边 $e : u \rightarrow v$，e 是必经边等价于 $Z = P_uQ_v$
- 路径条数太大, 要写高精?
  - 取个模就好了

整数拆分 Problem

- 给定 n 个正整数 $a_1, a_2, \ldots, a_n$
- 有 Q 个询问，每次给定 T ，询问 $\sum_{i=1}^{n} a_i x_i = T$ 是否有非负整数解
-
- $n \leq 100, \max a_i \leq 50000, Q \leq 10^5, T \leq 10^9$

整数拆分 Solution

- 令 $b = \min a_i$
- 若 T 有解，则 $T + b$ 也一定有解

整数拆分 Solution

- 令 $b = \min a_i$
- 若 T 有解，则 $T + b$ 也一定有解
- 令 d_i 表示：所有 $T \equiv i \mod b$ 且有解的 T 中，最小的 T 是多少
- 对于 d_i ，我们有 $d_{(i+a_t) \mod b} \leq d_i + a_t$

整数拆分 Solution

- 令 $b = \min a_i$
- 若 T 有解，则 $T + b$ 也一定有解
- 令 d_i 表示：所有 $T \equiv i \mod b$ 且有解的 T 中, 最小的 T 是多少
- 对于 d_i ，我们有 $d_{(i+a_t) \mod b} \leq d_i + a_t$
- 构图：对于每个 i, t ，连边 $i \to (i + a_t) \mod b$ ，权值为 a_t ，求最短路
  - 初始条件：$d_0 = 0$ ，即求从 0 到各点的最短路
  - d_i 即为到点 i 的最短路

基础知识

- 定义：权值和最小的生成树
-
- 主要算法思想: 安全边算法
  - 每次添加一条不会形成环的权值最小的边

最小生成树的性质

- 在一个权值

环性质

任意一个环上，权值最大的边一定不在 MST 上。

割性质

任意一个割中，权值最小的边一定在 MST 上。

Kruscal 算法

- 基本流程
  1. 把边按照边权从小到大排序
  2. 用并查集维护连通性
  3. 从小到大枚举每条边，加进去不构成环就加进去
- 时间复杂度 $O(E\log E)$
- 优点：速度快，好写
  - 通常来说图的边数会比较大

Prim 算法

- 基本流程
  1. 选择任意一个点
  2. 每次选择到被选择的点距离最小的点加入
  3. 维护每个还没有被选择的点到被选择的点的最小边权
  4. 注意和 Dijkstra 算法比较
- 正确性：割性质
- 时间复杂度 $O(V^2)$
  - 可以用堆优化到 $O((E+V)\log V)$

最优比率生成树 Problem

- 给定一个图 $G = (V, E)$，每条边有两个权 x_e, y_e
- 求一棵生成树 T，使得 $f(T) = \dfrac{\sum_{e \in T} x_e}{\sum_{e \in T} y_e}$ 最大
-
- $|V| \le 10^5$,$|E| \le 10^5, x_e, y_e \le 10^9$

最优比率生成树 Solution

- 经典 0/1 分数规划问题
- 二分答案 k ：是否存在一棵生成树 T 使得 $f(T) \geq k$ ？
- $$f(T) \geq k \Leftrightarrow \sum_{e \in T} x_e \geq k \sum_{e \in T} y_e \Leftrightarrow \sum_{e} (x_e - k y_e) \geq 0$$

最优比率生成树 Solution

- 经典 0/1 分数规划问题
- 二分答案 k ：是否存在一棵生成树 T 使得 $f(T) \geq k$ ？
- $$f(T) \geq k \Leftrightarrow \sum_{e \in T} x_e \geq k \sum_{e \in T} y_e \Leftrightarrow \sum_{e} (x_e - k y_e) \geq 0$$
- 令每条边的边权为 $x_e - k y_e$ ，求 MST 看权值和是否不小于 0

次小生成树 Problem

- 给定一个图 $G = (V, E)$
- 求一棵非 MST 的生成树，使得这棵生成树的权值最小
-
- $n \leq 10^3$,$m \leq 10^5$

次小生成树 Solution

Theorem (结论)

最小生成树与次小生成树只差一条边。

- DFS 求出 MST 上任意两点之间路径的最大边权
- 枚举每条非树边，尝试用这条边的边权替代边权最大的边来更新答案

权值生成树 Problem

- 给定一个图 $G = (V, E)$
- 求这个图的一棵生成树，使得 $\sum\limits_{v \neq 1} w_i s_i$ 最小
- 其中 w_i 表示以一为根时 i 与其父亲的边的权值，s_i 表示 i 的子树大小
-
- $|V| \leq 10^5, |E| \leq 3 \times 10^5, w_e > 0$

权值生成树 Solution

- 把原式变形 $\sum_{v \neq 1} w_i s_i = \sum_{v \neq 1} dist_v$
- 其中 $dist_v$ 表示 v 到 1 的距离

权值生成树 Solution

- 把原式变形 $\sum_{v \neq 1} w_i s_i = \sum_{v \neq 1} dist_v$
- 其中 $dist_v$ 表示 v 到 1 的距离
- 求每个点的最短路，最短路树即为答案
- 所以这个题其实是和 MST 无关的

最小极差生成树 Problem

- 给定一个图 $G = (V, E)$
- 求一棵 G 的生成树 T，使得 T 中边权的极差最小
- $|V| \leq |E| \leq 5000$

最小极差生成树 Solution

- 将边按照边权排序
- 枚举最小边，从这条边开始，不断加边，直至图连通
- 已经加入的边中一定可以找到一棵生成树
- 并且最后一条加入的边一定是所有可能的最大边中，最小的一个
- 复杂度 $O(nm)$
  - 如何继续降低复杂度？

最小乘积生成树 Problem

- 给定一个图 $G=(V,E)$，边有两种权值 $x_e,y_e>0$
- 求一棵生成树 T 满足
$$\sum_{e\in T} x_e \times \sum_{e\in T} y_e$$
最小
-
- $|V|\le 50, |E|\le 200, x_e,y_e\le 50$

最小乘积生成树 Solution

- 对于每棵生成树 T ，可以得到一个二元组 $(X = \sum_{e \in T} x_e, Y = \sum_{e \in T} y_e)$
- 把所有的二元组画在平面上，可以得到平面上面的若干个点
    - 最优解一定在凸包上！

最小乘积生成树 Solution

- 对于每棵生成树 T ，可以得到一个二元组 $(X = \sum_{e \in T} x_e, Y = \sum_{e \in T} y_e)$
- 把所有的二元组画在平面上，可以得到平面上面的若干个点
  - 最优解一定在凸包上！
- 如何找到凸包：Quick-Hull
  - 找到凸包上任意两个点，作一根线，找到此斜率下最优的点
  - 分治

最小乘积生成树 Solution

- 对于每棵生成树 T ，可以得到一个二元组 $(X=\sum_{e\in T}x_e, Y=\sum_{e\in T}y_e)$
- 把所有的二元组画在平面上，可以得到平面上面的若干个点
  - 最优解一定在凸包上！
- 如何找到凸包：Quick-Hull
  - 找到凸包上任意两个点，作一根线，找到此斜率下最优的点
  - 分治
- 如何找到一个斜率下最优的点？
  - 最小化 $X-kY=\sum_{e\in T}(x_e-ky_e)$
- 复杂度：与分治次数相关，而分治次数最多不超过凸包上的点数
  - 注意每个点横坐标均为整数，且被 $n\times\max x_e$ 限制住

瓶颈路径 Problem

Definition (瓶颈路径)

两点间的瓶颈路径即为所有连接两点的简单路径中，最大边最小的一条路径。

- 给定一个图 $G = (V, E)$
- 有 Q 个询问，每次询问两点间瓶颈路径的权。
-
- $|V| \leq 10^3, |E| \leq 3 \times 10^5, Q \leq 10^3$

瓶颈路径 Solution

Theorem (环切性质)

在图 $G=(V,E)$ ，如果存在一个环，把环中权最大的边 e 删除得到图 G' ，则 G 和 G' 的最小生成树权和相同。

- MST 上 S 到 T 的路径一定是 $S \to T$ 的瓶颈路径
- 暴力查询树上两点间最大边权即可
  - 各种各样的方法均可以做到 $O(\log V)$ 每次查询

连通国家 Problem

- 某个国家有 n 个城市
- 已知每个城市的坐标，两点间的距离为欧几里得距离
- 现在想把所有城市连通起来
- 两个城市 A, B 连通，当且仅当以下条件中至少有一个得到满足
  - 有一条连接 A, B 的公路
  - A, B 都建有飞机场
  - 存在一个城市 C 使得 AC 连通，BC 连通
  - 建造单位长度公路的代价为 x，修建一个飞机场的代价为 y
- 求最小总代价
-
- $n \leq 10^3$

连通国家 Solution

1. 没有城市建造飞机场
   - 就是最小生成树
2. 有城市建造飞机场
   - 新建一个超级点，所有点与这个点的边的费用均为 y
   - 一个城市建立飞机场 $\Leftrightarrow$ 这个城市与超级点的边在生成树中
   - 求新图的 MST 即可

概念、算法

- 序列 S 是图 G 的一个拓扑序列，当且仅当:
  - 每个顶点出现且只出现一次
  - 若 A 在序列中排在 B 的前面，则在 G 中不存在从 B 到 A 的路径
- 如何求拓扑序列?
  - 记录每个点的入度
  - 一旦入度为 0 就加入队列
  - 线性

DAG 判断 Problem

- 给定一个有向图 $G = (V, E)$
- 判断这个图是否存在环
-
- $|V| \leq 10^5, |E| \leq 3 \times 10^5$

DAG 判断 Solution

- 一个图是 DAG 当且仅当它存在拓扑序列
- 尝试对这个图进行拓扑排序
- 如果某时刻还有点没有被访问到但是队列为空了，则这个图存在环

拓扑序 Problem

- 给定一个有向无环图
- 要求一个拓扑序，使得 1 在拓扑序中的位置尽量考前，其次再是 2 的位置，再是 3 的位置，等等
- 例如 3 个点一条 3 到 1 的边，所求拓扑序是 3 1 2
- $n \leq 10^3$

拓扑序 Solution

- 考虑拓扑序的最后一个点
  - 一定没有出度
- 贪心：这个点一定是所有没有出度的点中标号最大的一个
- 每次选择没有出度的点中，标号最大的一个，然后把它从原图中删除
- 答案即为这个序列翻转后的序列

欧拉回路问题 Problem

Definition (欧拉回路)

图 G 的一个回路，若它恰通过 G 中每条边一次，则称该回路为欧拉（Euler）回路。

- 给定一个有向图 $G = (V, E)$，求它的一条欧拉回路
-
- $|V| \leq 10^5, |E| \leq 3 \times 10^5$

欧拉回路问题 Solution

- 存在欧拉回路的充要条件
  - 基图连通
  - 每个点的出度等于入度
- 证明：数学归纳法
  - 一定存在环
  - 将环删去，得到若干个联通块
  - 联通块可由环串起来
- DFS 每个点，出栈序列即为欧拉回路

最大团 Problem

Definition (团)

对于无向图 $G = (V, E)$，V 的一个子集 S 被成为团，当且仅当对于 S 中任意两个点 x, y，均存在一条连接 x, y 的边。

- 大小最大的团即为最大团
- 给定一个图 $G = (V, E)$，求最大团及其数目
-
- $|V| \le 50, |E| \le 200$

最大团 Solution

- 对于一般图来说，最大团是 NPC 的。二分图？不知道
  - 所以此题没有多项式算法
- 搜索 $+$ 剪枝
- 令 f_i 表示前 i 个点所组成的图的最大团大小
- 在搜索第 $i+1$ 个点的时候，如果 $f_i + cur < best$ 则不需要继续搜下去了
  - cur 表示当前搜索过程中已经选了多少个点
  - $best$ 表示当前搜索过程中的最优解
- 依次求 f_i 即可

树的直径 Problem

- 给定一个边带正权的树 $T = (V, E)$，求任意两点之间最短路的最大值，即

$$\max_{s,t} dist_{s,t}$$

-
- $n \leq 10^5$

树的直径 Solution

- 对于任意一个点 x，从 x 开始 BFS 一次找到的最远点 y 一定是某条直径的一个端点

Proof.

- 不妨令直径为 s 到 t
- 考虑 $s — t$ 与 $x — y$ 的相交情况
  - 不相交
  - 相交于一点
  - 相交于若干段

$\square$

- 两次 BFS 即可

TC Problem

- 已知一个无向图 $G = (V, E)$ 恰好包含 m 个团
  - 即，G 中任意一条边都属于 m 个团中的某一个
- 给定这 m 个团
- 求

$$\prod_{s<t} dist_{s,t} \mod 10^9 + 7$$

  - 单位边权
-
- $n, m \leq 2000$ ，给定的团的大小总和不超过 10^4

TC Solution

- Floyd：复杂度太高
- 对于每个点开始 BFS ：边数过大
  - 如何减少边数？
- 加点，把团变为星型
  - 若 $v_1, v_2, \ldots, v_k$ 是一个团
  - 则加一个点 u ，连边 $u \leftrightarrow v_i$ ，边权为 0.5
- $dist_{i,j}$ 不变，边数降为 10^4 ，点数为 $n+m$
  - 对于每个点开始 BFS 求最短路即可