

二十棵树问题

数学研发论坛 Wayne 编辑整理

0 问题导读

这里所说的 20 棵树问题,通俗的说,就是 20 棵树,每行 4 棵,最多能种多少行?

严格的来阐述,就是^①: 欧氏平面上的 n 个点,若其中任 $m+1$ 个点不共线,问恰过其中 m 个点的直线最多有几条? ($n>m$) 当 $n=20$, $m=4$ 时,就是 20 棵树植树问题.

事实上,20 棵树问题是 Sylvester 在 1867 年提出的果园问题中最引人关注的一部分. 至今该问题还频频出现在中小學生所涉猎的数学课外读物中.

1 国内外研究现状

然而该问题并非大家所想象的有点难.而是远超出一般的难度.

早在十六世纪,古希腊、古罗马、古埃及等都先后完成了十六行的排列并将美丽的图谱广泛应用于高雅装饰建筑、华丽工艺美术(图 1)。进入十八世纪,德国数学家高斯猜想 20 棵树植树问题应能达到十八行,但一直未能见其发表绘制出的十八行图谱。直到十九世纪,此猜想才被美国的娱乐数学大师山姆·劳埃德完成并绘制出了精美的十八行图谱,而后还制成娱乐棋盛行于欧美,颇受人们喜爱.

二十世纪七十年代,两位数学爱好者巧妙地运用电子计算机成功地绘制出了精湛美丽(由三个五角星构成)的 20 行图.



于是所有人都以为最多能排 20 行,至今的网络上依然充斥着最多 20 行的解答.

1985 年,我国的伍毛成功地绘制出了 21 行植树图.

2005 年, Lin Xianzu (我想应是中国人) 证明了至少存在 23 行的植树法, 并给出了在射影平面上的 23 行图.

2006 年, 辽宁锦州开发区笔架山小学王兴君成功地绘制出了欧氏平面中 23 行图.

2009 年, 河北省邢台学院学生黄阳阳 (数学研发论坛的昵称 eyond) 成功地绘制出了另一个漂亮的 23 行植树图.

2009 年, 在本论坛, 在 mathe, 无心人, zgg, sheng_jianguo, 数学星空等一堆爱好者的持续数月的讨论下, 取得了重大的突破. 不仅确定了最多能排 23 行, 而且给出了所有可能的解. 神奇吧!

2 最多能排二十三行

果树问题在本论坛讨论的破冰之贴是 <果树问题讨论:这两个问题等价么? >, 该帖作者企图通过解决其等价问题来间接解决 20 棵树问题:

20 个字母 A-T. 取出 x 组, 每组 4 个字母, 每个字母在同一组内只能出现一次, 但可以同时包含在多个组内, 所选取的 x 组中任意两组中至多有一个字母相同, 求 x 的最大值.

然而这个命题跟 20 棵树问题是不等价的. 引用 635 楼 hujunhua 言简意赅的解释如下:

问题一 (即果树问题) 属于实射影几何, 问题二属于有限射影几何. 实射影几何与有限射影几何一个本质区别是: 前者遵守 Desargues 公理, 后者不遵守. 这个区别导致这样一个结果: 在有限射影几何中的共直线的 3 点, 在实射影几何中不一定找到模型。

但问题二相对来说, 比较容易通过计算机程序来实现. 理论上, 问题一最多能排的行数不大于 26, 问题二最多有 30 行. sheng_jianguo 也给出了问题二的一个 30 行的解. 可惜不是问题一的解.

{1,2,3,4},{1,5,6,7},{1,8,9,10},{1,11,12,13},{1,14,15,16},{1,17,18,19},
{2,5,8,20},{2,6,11,15},{2,7,14,19},{2,9,13,17},{2,10,12,16},{3,5,12,14},
{3,6,9,20},{3,7,13,18},{3,8,11,19},{3,10,15,17},{4,5,16,17},{4,6,12,19},
{4,7,10,20},{4,8,13,15},{4,9,14,18},{5,9,15,19},{5,10,11,18},{6,8,16,18},
{6,10,13,14},{7,8,12,17},{7,9,11,16},{11,14,17,20},{12,15,18,20},{13,16,19,20}}

于是, 如何高效的在庞大的问题二的解空间里, 搜索出问题一的解, 就是问题的核心了. mathe 在 2 楼给出了高效的遴选算法, 引用原文如下:

i) 搜索算法, 主要是去搜索问题二的解的方法去搜索, 其中比较重要的一个的方法在 260#, 它极大的缩小了搜索范围。

ii) 问题一方案的判定问题. 实际上现在我并没有一个可以完全判断一个问题二方案是否存在对应问题一解的程序. 但是 51# 的方案可以淘汰大部分非法方案。

所以结合 i) 和 ii), 我们可以有比较好的方法来搜索问题一的解。

iii) 而在搜索过程中, 有很多问题一的方案方案实际上是等价的, 它们可以通过置换树的字母来相互转换, 对于这样的结果, 我们也不需要重复搜索它们, 为此, 我提供了一个程序将每个问题一的方案的字母表示进行标准化 (程序中 node_edge_set 模板类), 其主要思路就是如果两个点的度数不同, 那么它们必然不同类. 然后如果两条边所通过的点集类别不同, 那么边也不同类. 再然后如果两个点所过的边的集合类别不同, 这些点又不同类. 当然即使这样递归下去, 最后还可能有些不同的图无法区分, 这时, 我们就可以通过对某些点打上一个特殊的标签 (于是它可以和原先同类的点区分开), 最终得到一个所有点和边的非类都区分开的图. 但是标签打在不同的点上, 结果会不同, 这个就需要我们从所有这些结果中选择一个标准的 (字典排列顺序最小的一个)。

iv) 最后还有一个加强就是在 i) 的搜索中, 直接和 iii) 相结合, 那么整个搜索空间可以看成是一个有向无环图, 其中图中每个点是一个问题二的方案. 于是对于图中每个点, 它可以有多个父节点, 这个会导致一个点可以被通过多个父节点重复搜索到. 而这时, 我们可以对于每个搜索到的节点, 判断当前搜索路径是不是通过它的第一个父节点, 如果不是, 也把分支裁减 (另外由于重边的存在, 程序会对同时保存每个节点的所有孩子节点, 然后将重复的淘汰)

进展到这一步,大军在后,而唯有 mathe 拼杀在前.不过,有无心人一直跟进 mathe 的程序实现.期间大家曾一度想做成分布式计算的版本.

大约一年之后,mathe 想到了高效的过滤候选解的方法,见 357 楼②.大致意思就是,每搜索到一个候选解的时候,从三点共线扩充到四点共线.很容易计算出这种扩充的候选解是否是问题一的解.做一个标记,下次搜索的时候就不用重复计算这个候选解了.

经过这种思路过滤之后的候选解大大减少,文件大小降至 6G 左右. 把文件拆分成 30 份,在无心人,数学星空,sheng_jianguo, winxos,Frankenstein,liangbch 等众多网友的参与下,终于计算出了问题一的解,有 2 个,其格式如下:

```
i)print(ACEFDFHJBCIJBEHKADIKBFGMEJLMBDLNCGNODEGPFKNPAJOPAGLQCK
MQCHLRDMOREIQRANSFIOBPQSGHITABRTCDST);
```

```
solve([+4/3+1*E_Y,-1/2+1*T_X,+3+1*P_Y,+2+1*Q_Y,-5/2+1*Q_X,+1/2+1*R_X,-3+1*P_X,+3/2+1*L_X,-3/2+1*M_X,-2+1*R_Y,+1+1*O_Y,-2+1*L_Y,+2+1*M_Y,-3+1*N_X,+1+1*N_Y,-2+1*H_Y,-1+1*O_X,-1+1*S_X,-3+1*K_X,+1+1*A_Y,+2+1*K_Y,-3/2+1*G_X,-3/2+1*B_X,+1+1*G_Y,-1+1*T_Y,-1+1*S_Y],[E_Y,T_X,P_Y,Q_Y,Q_X,R_X,P_X,L_X,M_X,R_Y,O_Y,L_Y,M_Y,N_X,N_Y,H_Y,O_X,S_X,K_X,A_Y,K_Y,G_X,B_X,G_Y,T_Y,S_Y]);
```

```
print("A=(1,A_y,0) B_y=0 C=(1,0,0) D_x=0 D_y=1 E=(1,E_y,0) F=(0,1,0) H_x=0 I_x=1 I_y=0 J_x=0 J_y=0 ");
```

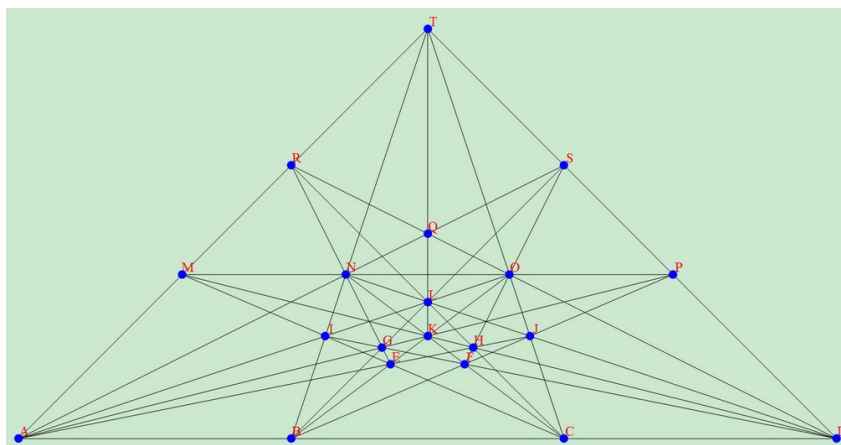
```
ii)print(ADGJBEIJCDHKAFIKCEGLBFHLCJMODINODLMPAHNPGKOPBGMQFJNQA
EOQEHRMBKNRCFPRIQRABCSDEFSGHITJKLTMNST);
```

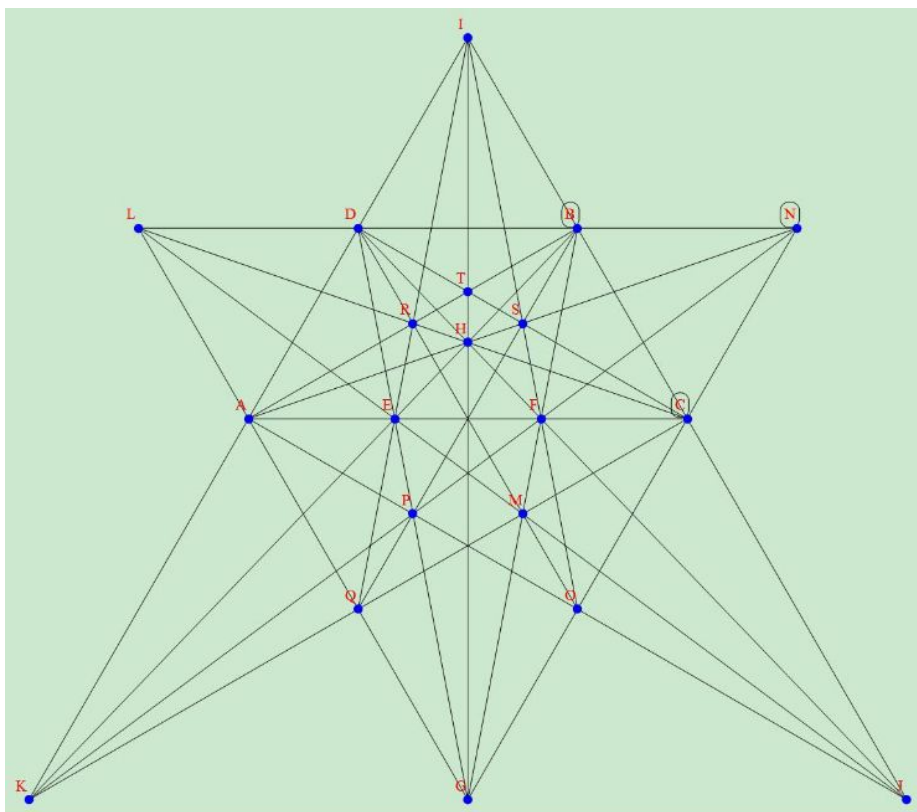
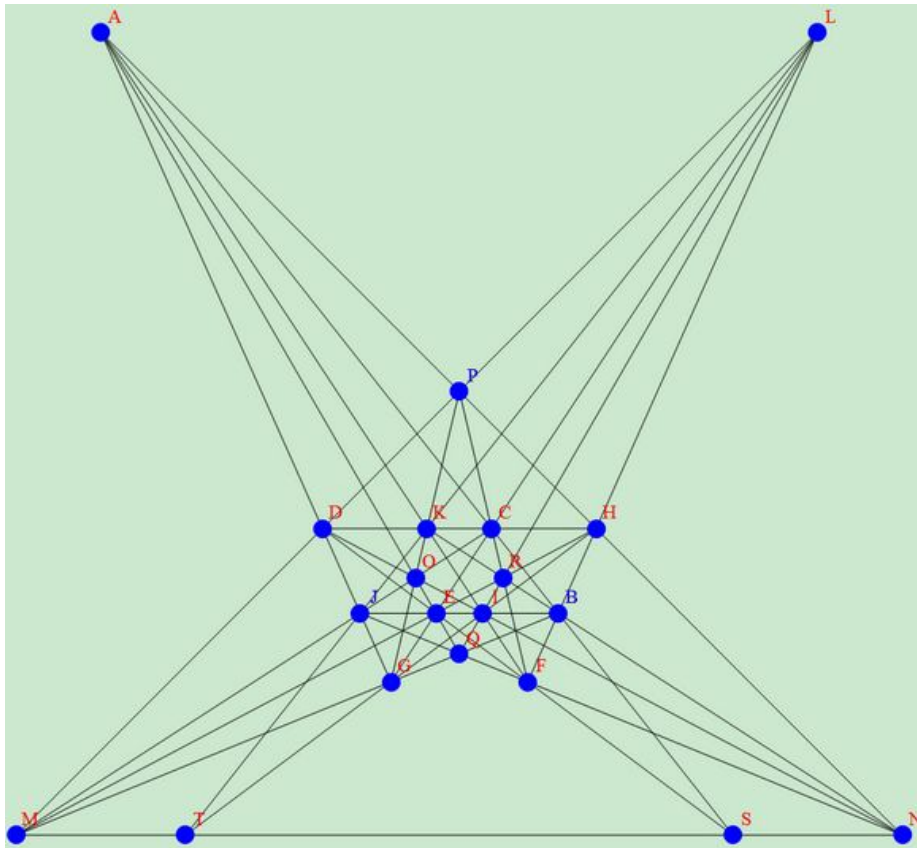
```
solve([+1+3*T_Y+1*T_Y*T_Y,-3/5+1*R_Y-2/5*T_Y,-4/5+1*R_X-1/5*T_Y,-3+1*H_X-1*T_Y,-4+1*P_X-1*T_Y,+1+1*S_X,-2+1*Q_X-1*T_Y,+2+1*O_X+1*T_Y,-1+1*L_Y-1*T_Y,+1+1*P_Y,-3+1*M_Y-2*T_Y,+1+1*D_Y+1*T_Y,-2+1*S_Y-1*T_Y,+1*G_Y-1*T_Y,-1+1*Q_Y,+1+1*N_Y,-2+1*C_Y-1*T_Y,-2+1*B_Y-1*T_Y,-1+1*O_Y,+1+1*H_Y,+2+1*C_X+1*T_Y,-2+1*N_X-1*T_Y,-1+1*L_X,-1+1*T_X,+2+1*M_X+1*T_Y,-2+1*F_X-1*T_Y],[T_Y,R_Y,R_X,H_X,P_X,S_X,Q_X,O_X,L_Y,P_Y,M_Y,D_Y,S_Y,G_Y,Q_Y,N_Y,C_Y,B_Y,O_Y,H_Y,C_X,N_X,L_X,T_X,M_X,F_X]);
```

```
print("A=(1,0,0) B_x=0 D=(1,D_y,0) E_x=0 E_y=1 F_y=0 G=(1,G_y,0) I_x=0 I_y=0 J=(0,1,0) K_x=1 K_y=0 ");
```

3 最优解的作图展示

先看看最优 23 行的解的图案吧:

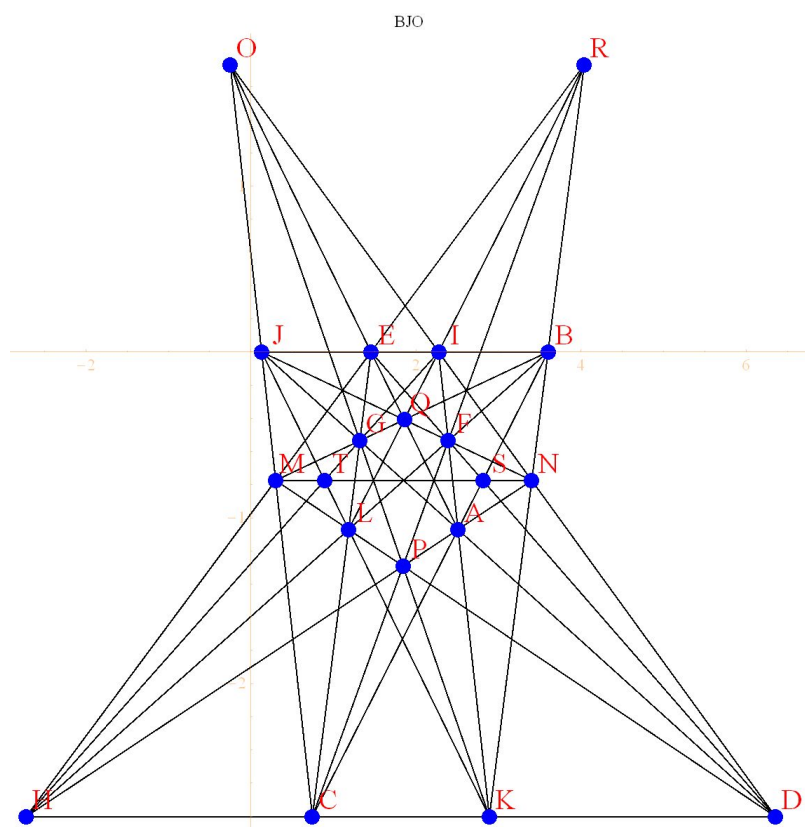
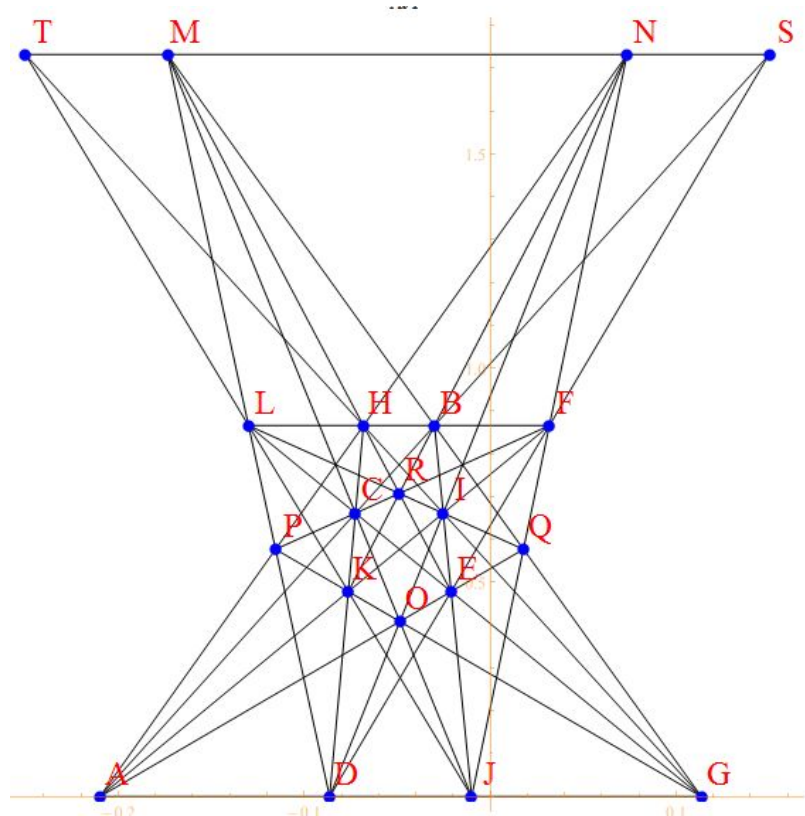




不是说只有两个解吗,怎么有三个图案呢?

事实上,上面程序得出的两个解是射影平面内的两个解. `mathe` 分析过,前面提及的王兴君和黄阳阳的两种解在射影平面内是等价的.即他们的点线结合关系是一致的.

更多不同的图案,请参考专帖 <果树种植最优解精美图形作法探讨>® .



4 参考链接

- ① [**果树问题讨论:这两个问题等价么？第 621 楼**](#)
- ② [果树问题讨论:这两个问题等价么？第 357 楼](#)
- ③ [果树种植最优解精美图形作法探讨](#)