

# 简况

[比赛链接](#)

AC 8题，Rank 49th

## 总结与反思

### cmx

写东西胆子还是大一些，这场的H笛卡尔树几乎是个裸题，可惜信心不足，要是写出来了就幸福了。另外D其实是想到一个更简单搞法的，不知为啥脑抽了。

### Ipy

### XSY

## 题解

### D. Points Construction Problem

这个题赛场上是Ipy通过基本图形组合的方式AC的。其实赛场上有人想过另一种方法，可惜脑抽了以为有情况没有覆盖。之后才发现是正确解法。

首先问题等效于构造n个小方块拼接成一个周长为m的图边形。显然m为奇数，是不行的。我们如果用m周长，构造一个尽可能大的矩形 $(m/2-1, m/2+1)$ 这样，最少我们可以沿着对角线填，个数是长边的长度。最多可以填出一个面积。可以发现，这样的矩形所能达到的方块个数的范围，是比其他矩形只多不少的。而且很容易证明，在这个范围之外，是不可能的。

当然有个问题 $[m/2-1, m/2+1]$ 之间所有方块数目都能取到吗？其实是可以的，往对角线两侧紧密填充即可，注意不要让图形出现凹陷，这样周长会大于m

by cmx

## 补题

## H.Sort the Strings Revision

笛卡尔树题。我们知道如果第一个字符替换出一个更小的字符，那么这个替换之后的操作得到的字符串都会在前面；更大则在后面，相等则等于没有任何影响，对第二个、第三个字符类似递归讨论。可以看出左右的递归形成了一棵二叉树。直接对p数组构造笛卡尔树，就是这个二叉树了。因为笛卡尔树可以选择相同的最小值取左为根。所以对于相等替换，我们全部对对应p修改为正无穷1E9即可。

建树完毕之后dfs[]根据大于还是小于分类讨论先左边还是先右边。得到的dfs序列就是我们最后的字符串字典序。

全递归版本也不到一秒，不用卡常。

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team



Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:2020\\_nowcoder\\_multiuniversity\\_3&rev=1595303459](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:2020_nowcoder_multiuniversity_3&rev=1595303459)

Last update: 2020/07/21 11:50