# Day 2 solution

acpty

September 20, 2019

# 前言

我太南了

## 华灵「蝶妄想」

这是一道比较简单的构造题
部分分暗示了各种情况，应该不会有人忘记讨论了吧

首先，括号序的长度必须是偶数
所以当n为奇数或m为奇数的时候直接把另一边全部用括号序填充就好了
大概是这样子：
n=3 m=4
(())
(())
(())

然后，注意到所有括号序的开头均为'('，结尾均为')'
也就是说第一行第一列最后一行第一列最后一列均不可能形成完整括号序
那么我们考虑构造这样一个东西：
n=6 m=6
.((((.
(()())
()()()
(()())
()()()
.)))).
这样的话答案是n+m-4

当然，在n,m较小的时候有更优的填法，比如样例中的n=2,m=2
我们考虑牺牲一半的行或列，这样构造:
n=4,m=6
((((((
()()()
)()()(
))))))
这样的答案是n/2+m-1
综合上面两种构造方法取较优的那一种即可
复杂度$O(nm)$

# 樱符「完全墨染的樱花」

这是一道中等题
如果能够发现题目中的条件的本质那么就会变得很简单

首先，我们可以发现，当边权为1时，任意两点最大流$\leq 2$，等价于这张图是仙人掌

Proof:

考虑反证，假设边(u,v)同时存在于两个简单环中

由于最大流=最小割，将点u和点v分开至少需要把两个简单环都断开并把(u,v)断开，即$maxflow(u,v) \geq 3$，与题设矛盾

既然每一条边最多存在于一个简单环中，考虑新图任意两点最大流，也就是最小割
显然最小割要么割掉一条桥边，要么割掉同一个环里的两条边
考虑如果割掉同一个环里的两条边，无论怎样环中边权最小的那条边一定会被割掉
那么我们可以对于每个环，将其边权最小的边删去，剩下的边的边权加上删去的边的边权，这样操作之后任意两点的最大流不会改变

容易发现这样操作后我们的图变成了一棵树
对于一棵树，任意两点最小割等价于路径最小值
把所有边按权值从大到小排序依次加入，用并查集维护每个连通块的$\sum p^{i(n-1)}$和$\sum p^i$即可
复杂度$O(n\log n)$

Source: [CodeChef MAXDTREE]
这是一道较难的题
首先需要想到办法判定一个数是否存在于序列a中
然后再用Dp套Dp的方法来求出答案
根据状态可能还需要前缀和优化

# 幽曲「埋骨于弘川」

首先考虑如何判定一个数存在于a中

由于每次加的数为[1,k-1]，最多产生1的进位，一定有一个时刻，当前的数形如xxxxxxxxx0000000000a

即个位是一个数，然后往前大段都是0

我们规定个位为1位，位数越高越大

考虑设状态f[i][p][a]表示，第2到i位都为0，个位为a，高位的最大值为p，要在第i+1位产生1的进位，a会变成多少

f[i][p][a]可以用O(k)从f[i-1][p'][a']转移过来，相当于在i-1位做了k次进位

然后再设g[i][p][x][a]表示，高位已近填完了，前面的最大值为p，个位为a，要在第i为填x的话a会变成多少
这个可以也一样用f[i-1][p'][a']做k次进位转移来
预处理的复杂度为$O(nk^3)$

判定一个数的话，一开始x为1，p为0，然后从高往低位，不断的x=g[i][p][d[i]][x],p=max(p,d[i])
其中d[i]表示这个数第i位的位值
只要这个过程能够合法，那么这个数就能被表示出来

我们可以把这个判定的过程放到Dp里
设Dp[i][j][p][x]表示，当前做到点i，判定到第j位，最大值为p，个位为x的方案数
枚举下一个点k，Dp[i][j][p][x]可以转移到Dp[k][j-1][max(p,d[k])][g[j-1][p][d[k]][x]]
这样做的复杂度是$O(n^3k^2)$的

## 幽曲「埋骨于弘川」

其实我们不妨按dfs序转移
考虑能够转移到点i的点，那便是dfs序在i的父亲后，在i前面的所有点，也就是一段区间
用前缀和优化即可
复杂度$O(n^2k^2)$