NOIp 2017 solution

Dez

2017 年 12 月 4 日

1 Math

最终结论是 $ab-a-b(a,b\in N_+)$ 是最大的 a,b 不能表示出来的正整数。

要证明这个结论，首先要证明对于不定方程 $ax+by=c(a,b,c>0)$，存在一组解 (x_1,y_1)，使得 $0<x_1,-a<y_1\leq 0$。因为我们可以求出一组特解 (x_0,y_0)。那么通解的形式就是 $(x_0+bt,y_0-at)(t\in Z)$。这样就一定能找到一个 y 满足 $-a<y\leq 0$。这时 x 毫无疑问是正数。

然后再来看我们要证明的，先假设 $ab-a-b$ 可以被表示，那么有 $ab-a-b=ua+vb$。

那么 $ab=(u+1)a+(v+1)b$。然后可以得到 $a\mid(v+1)$。

令 $ka=v+1$，带回上式得 $ab=(u+1)a+kab$。

则 $(u+1)a=(1-k)ab$，而 $(1-k)$ 都不是个正数，矛盾。

所以证得 $ab-a-b$ 不能用 a,b 表示。

接下来证明形如 $ab-a-b+t(t\in N_+)$ 的数可以被 a,b 表示出来。

即证明 $ab-a-b+t=au+bv$，其中所有数都是非负整数。

使得 $ax+by=t$，原式变为 $ab-a-b+ax+by$。

即 $(x-1)a+(a-1+y)b$。

而我们先前已经证明了存在一组解让 $0<x_1,-a<y_1\leq 0$

那么此时 $x-1\geq 0,a+y-1\geq 0$，原式得证。

我还是没搞清为什么我那个式子是对的。（猜想它恰好是那组特解，但是不知道和答案有什么关联）

2 Complexity

模拟题。依我看就是考你条理够不够清晰，还有一点概念的掌握。只要代码写得清楚就可以了。不过还是花了我蛮多时间。

3 Park

一个动态规划。学了一个比较简洁的记忆化搜索。大致思想是先从终点出发做最短路，然后从起点开始搜索到终点的路径。记 $f(x,k)$ 为在点 x 处到终点与最短路相差 k 的路径有多少条。为什么要从终点求最短路呢? 这样在搜索的时候就不会搜到到不了终点的路上，从一开始就减少了非常多的无用状态。这是这个算法高效的原因。我认为大致的时间复杂度是 $O(NK)$。

4 Cheese

只要暴力搜索判断连通就行了。以下边界为起点放入队列，每次取队列头把所有可以连通的圆都加进队列。搜到上边界或者不能再扩展为止。这题精度要求较高，需要精确到 10^{-8} 才能通过全部数据。时间复杂度 $O(n^2)$。

5 Treasure

枚举子集的动归。以前没有接触这一方面的题，算是趁着这个机会了解一下。枚举子集最不同的一个地方在于它的每一个状态从不同的地方转移得到的不一定是最优解。但是这种做法又可以保证存在一种转移方式构造出最优解。至于这道题，我们首先预处理出每个点到每个集合的最小花费。具体做法就是枚举每个点和每个集合，然后暴力求出最小的花费。然后就是枚举子集了。我们在这里设 $f(d,s)$ 为当前的深度为 d，连通状态为 s 的时候的最小花费。每枚举出一个子集，就利用我们的预处理求出它的补集中所有的点在深度为 $d-1$ 时连到子集的花费和，那么 $f(d,s)=f(d-1,u)+(d-1)\times c_{vu}$。其中 $u\subset s, v\subset s$，且 v 是 u 的补集。将初始状态之外的状态设为无穷大即可。时间复杂度为 $O(n3^n)$。(证明需要二项式定理)

6 Phanlax

一个比较难码的数据结构题。思路就是每一行用一个平衡树维护（实际上存的是一些区间，叫区间树比较合适），每分裂一次就加进新点，把原点拆成两个。然后最后一行单独用一个平衡树维护，这个里面不仅要删除还要不断插入。需要写带区间分裂与合并的平衡树。代码量完全取决于实现的优美程度。并且线段树也可以解决这个题。注意编号会超出 int 类型的范围。时间复杂度 $O(n \log n)$。（常数巨大！）