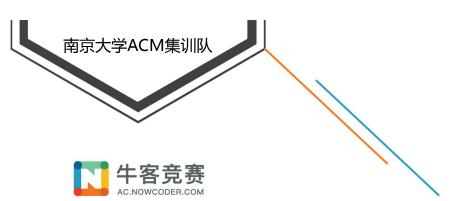


2019牛客暑期多校训练营 第十场



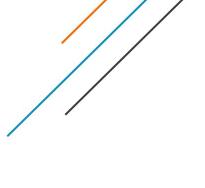


Problem H: Stammering Chemists

• easy , 签到 , 模拟

方法一: 按照题意(图片)将每种形态的树按照特征(度数,叶子等)区分。

方法二: 对每棵树编码, 枚举每个全排列, 判断是否完全相同。







Problem B: Coffee Chicken

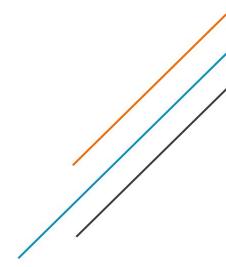
· easy, 签到, 递归

记f(i)表示S(i)的长度, f(i)=f(i-2)+f(i-1)。

递归调用solve(k,n)表示S(k)的第n个字符。

若k<=2直接返回答案。

k>2时, 若n>f(i-2), 调用solve(k-1,n-f(i-2)); 否则调用solve(k-2,n)。







Problem E: Hilbert Sort

• easy , 编码 , 排序

题目给出的是"希尔伯特曲线",属于"分形"。

给所有点编码,具体可以递归实现,用get(x,y,k)表示大小为2^k的希尔伯特曲线中(x,y)号点的坐标。

然后讨论(x,y)在四个角的情况,然后进行一定的坐标变换即可。





Problem D: Han Xin and His Troops

- · medium-easy,数论
- 判定给定方程组在整个自然数域无解的条件:存在(i,j), 令d=gcd(pi,pj), ai mod d ≠ bi mod d。
- 在有解的条件下,找出最小的满足条件的解有许多方法:
- 方法一: 先将所有同余方程组分解成对于所有模数都两两互质的形式, 然后套用中国剩余定理, 注意运算过程中超过long long的情况, 可以通过python或__int128辅助实现。
- 方法二:注意构造满足前i个方程的最小自然数解。假设前i-1个方程的最小自然数解为f(i-1), / lcm(p1,p2,...,pi-1)=y,则求解f(i)时,令f(i)=f(i-1),然后让f(i)每次加上y,直到满足第i个方程未知。可以证明,在保证有解的情况下,加法运算不会超过pi次。
 - std使用了方法二的做法,实现比较简单,不会用爆long long的问题,但是该方法仅限于求解pi比较小的 /同余方程组。



Problem F: Popping Balloons

- · medium-easy, 贪心, set
- 用f(i)表示中间一枪打第i行,能够射中的气球个数;用g(i)表示中间一枪打第i列,能射中的气球个数。
- 用multiset存所有g(i)的值,枚举中间一枪打第x行,将对每一个位于第x-r,x,x+r行的气球,将它们影响到的列(共三列)的g(j)的值更新,然后更新multiset内的元素。
- 中间一枪打第x行的最大收益即f(x)+(当前multiset内最大元素)。





Problem J: Wood Processing

- medium-easy , 动态规划
- 每次裁剪的一刀一定是对于高度连续的一段, 故先将所有木板按高度从大到小排序。
- dp[i][j]表示前i刀,割了高度前j高的木板,所浪费的最小代价。
- 转移有dp[i][j]=max_k(dp[i-1][k]+sum[j]-sum[k]-height[j]*(lens[j]-lens[k])), 其中sum[x]表示前x高的木板的总面积, lens[x]表示前x高的木板的总宽度。
- 上述转移拥有决策单调性,可以通过经典的分治优化dp进行优化。
- wqs二分在本题也适用。
- 复杂度O(nk logn)。





Problem G: Road Construction

- · medium , 几何
- 最优的直线一定平行或垂直于两个点的连线。
- 枚举最优直线斜率(最多有n^2个),然后用kth-element找到以这条直线排序的中间两个点,答案即两个点到该直线的距离差。
- 复杂度: O(n^3)。
- 本题可以通过斜率排序+旋转直线斜率的方法,做到O(1)更新所有点到直线的距离排序顺序(因为每次只交换相邻2点),可以将本题的复杂度优化到O(n^2 logn),但是由于本题重点在于考察发现直线的性质, / 故没有做加强。



Problem A : Blackjack

- · medium, 动态规划, 概率
- Calabash的最优策略即一直抽牌,直到抽到的牌的点数>a为止。
- 枚举Calabash最后一张抽的牌(设点数为x),若他抽完这张牌,点数和>a,且获胜(点数和<=b) 之前抽到牌的点数和在区间(a-x,min(a,b-x)]内。
- 用dp[i][j]表示除了枚举的牌,在剩余的牌中抽的前i张牌,点数和为j的概率。
- 转移:f[k][i][j]=f[k-1][i][j]+f[k-1][i-1][j-x_k]*i/(n-i+1)。dp[i][j]=f[n-1][i][j]。
- 则之前抽了i张牌,点数和j在(a-x,min(a,b-x)]内,对答案的贡献即dp[i][j]/(n-i)。





Problem A : Blackjack

- · medium, 动态规划, 概率
- 枚举一张牌后,计算对答案的贡献是O(500^2)的,但是每次计算dp值的复杂度为O(500^3)。总复杂度为O(500^4)。
- 这里可以利用"可逆背包"的做法,先预处理出F[i][j]表示考虑所有牌,抽到前i张牌,点数和为j的概率 然后每次枚举一张牌时,可以O(500^2)撤销这张牌对F[i][j]的贡献,O(500^2)的得到dp值。
- 总复杂度: O(500^3)。



Problem C: Gifted Composer

- · medium , 二分 , hash
- 考虑长度x作为周期出现的时间,一定是从时间t=x时,直到某个时刻的一个区间。
- 可以通过二分找到长度x作为周期消失的时间。
- 如何快速判断一个字符串S[1..n]是否有x作为周期?
- 只需判断S[1..n-x]是否等于S[x+1..n]即可。由于询问是离线的,可以预处理出整个最终串的Kash值,这样二分答案后的每次询问都可以O(1)回答。
- 复杂度: O(n logn)。





Problem I: Travel Dream

- hard, 随机, color-coding
- 每次将所有点随机染成0/1的颜色,假设最大环为(v1,v2,...,vk),则环上恰好连续k/2个点颜色为0,剩下k-k/2个点颜色为1的概率为k/(2^k)。
- 现在我们只要分别对颜色为0,1的点的连通快,用O((n+m)^2)的时间求出每两个点对之间长度为k/2-1和 (k-k/2)-1的最长简单路径长度,分别用f[0][u][v],f[1][u][v]存储。其中k/2-1,(k-k/2-1)<=4,
- 再枚举两条跨过0,1两个联通块的边e1(u1,v1),e2(u2,v2),不妨设u1,v1颜色为0,u2,v2颜色为1,有 f[0][u1][u2]+e1+e2+f1[v1][v2]更新答案即可。
- 每次随机后,最大答案将以k/(2^k)的概率得到。若随机T次,得不到最大答案的概率为(1-k/(2^k))^T。
- ✓k=10时,取T=1000,(1-k/(2^k))^T=5e-5,可以忽略不计。





Problem I: Travel Dream

- hard, 随机, color-coding
- 分类讨论,给出O((n+m)^2)找到一张图中任意两点之间长度为len=1,2,3,4的最长简单路径做法:
- len=1: 直接枚举边。复杂度: O(m)。
- len=2: 直接枚举两条边,判断是否共点,若共点(u,w),(v,w),则更新(u,v)的答案。
- len=3: 枚举两条边,判断是否共点,若不共点(u1,v1),(u2,v2),若(u1,u2)有边,更新(v1,v2)的答案, 对其余3个pair做类似的更新。
- len=4: 用len=2的方法,先对每个点对(u,v)求出长度为2的前3长的路径已经对应的经过点编号。然后枚举两条边,判断是否共点,若不共点(u1,v1),(u2,v2),取(u1,u2)之间最长的且不经过v1或v2的长度为2的路/径,更新(v1,v2)的答案,对其余3个pair做类似的更新。



Thanks

