2017福建省夏令营Day2练习-solution

by zld3794955

2017 年 7 月 18 日

题目名称	奇怪的道路	黑白矩阵	商店
目录	road	table	shop
可执行文件名	road	table	shop
输入文件名	road.in	table.in	shop.in
输出文件名	road.out	table.out	shop.out
每个测试点时限	1秒	1秒	1秒
内存限制	256MB	256MB	256MB
测试点数量	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

注意：预计是Win8下Cena评测。

评测机性能足够好且评测时开启了-O2优化

评测时栈空间设为32M，足以满足正常的函数递归需求。

奇怪的道路

【算法思路】

样例已经在提醒我们，只要房屋沿道路顺序编号，则两个房屋之间道路的长度即为两个房屋的编号之差，则我们只要能快速求出任意一个位置房屋的编号即可。

Easy难度直接在(c)图上数数打表即可。

Normal难度可以写代码模拟城市生成过程，打出城市中每个位置对应的编号表。

而Hard难度的话需要观察，注意到这样的道路设计有着非常明显的相似子结构，因此我们考虑用分治类算法处理。

设$getnum(n,x,y)$为阶为n的城市中第x行第y列的房屋的编号，那么当$n=0$时，其结果为1，否则，我们通过对图形的观察可知：

当(x,y)在左上部分时，结果即为$0\times 2^{2n-2}+getnum(n-1,y,x)$。

当(x,y)在右上部分时，结果即为$1\times 2^{2n-2}+getnum(n-1,x,y-2^{n-1})$。

当(x,y)在右下部分时，结果即为$2\times 2^{2n-2}+getnum(n-1,x-2^{n-1},y-2^{n-1})$。

当(x,y)在左下部分时，结果即为$3\times 2^{2n-2}+getnum(n-1,2^{n-1}+1-y,2^n+1-x)$。

这样我们就圆满地解决了这个问题。

黑白矩阵

【算法思路】

Easy：

直接利用上午讲过的递归回溯框架进行枚举，并利用题目中的提示对解的合法性进行判断即可，时间复杂度$O(nm \times 2^{nm})$。

Normal：

稍微掌握一点儿DP知识的同学应该能看出Normal难度能用状压DP搞定，当然，根据题目中的提示，我们实际上只要确定第一行和第一列的格子的颜色便能确定整张表格中其它格子应该填什么颜色，那么我们只需要直接递归回溯枚举第一行和第一列格子的颜色便能推出整张表格，但这里数据组数有点多，我觉得直接暴力搜索是过不了的（没写过，也许实际上能过，数据比较水），如果进行了一定的剪枝那么应该是很稳的。

而事实上，题目条件等价于每行的格子颜色均与第一行完全相同或完全相反（对列同理），那么我们只要枚举第一行的格子（也可以是第一列），便能用贪心求出这种情况下最少需要改变多少个格子的颜色，单组数据时间复杂度$O(2^m \times nm)$。

Hard：我们考虑题目中k这个奇怪的条件，这个条件告诉我们，如果结果不是-1的话，那么至多有k行存在被修改的格子，而剩下$n-k$行的格子均没有变化过。

因此，若$n > k$，则我们可以枚举哪一行的格子没有被修改过并以此进行贪心，显然，在枚举$k+1$个不同的行之后，如果存在合法的最优解，那么一定会被我们枚举到，单组数据时间复杂度$O(knm)$，若$n \leq k$，则我们直接枚举第一列的格子颜色，单组数据时间复杂度不超过$O(km2^k)$。

这样问题就被完美地解决了。

商店

【算法思路】

这题的Easy难度直接用long long贪心即可。

Normal难度可考虑建优先队列来获取当前最便宜的商品，但是这时候建议对商品价格取一下对数再进行比较，不然会不会爆double我也不是很清楚呢……

Hard难度的话，注意到我们都是取最便宜的n件商品，所以我们可以理论上二分答案ans，判定方法即为统计一下不超过ans最多可以买几件商品。

但是这种理论上的二分显然过不了这道题，我们需要压缩一下二分价格的范围。

先考虑枚举我们买了多少件第k种的商品，设为t件，接下来我们二分一下买了$t+1$件的商品的种数，由这些信息便可以确定最贵的物品的价格，相关的统计也十分简单，$O(\log k)$做一遍就出来了

显然，由于商品价格指数级增长的原因，在最终的方案里，初始价格为k的物品与初始价格为1的物品购买的种数不会相差太多，所以我们只需要在n/k这个值的附近枚举变量t即可。

总的时间复杂度$O(T\log^2 k)$。