牛客暑期ACM多校训练营

第 2 场-laofudasuan

牛客网
NOWCODER

A run

题目大意

白云在健身，每秒可以走1米或跑k米，并且不能连续两秒都在跑。
当它的移动距离在[L,R]之间时，可以选择结束锻炼。
问有多少种方案结束。
10^5组询问，所有询问的k都一样，$L, R \leq 10^5$

做法

DP?
f[i][0/1]表示已经跑了i米，最后一步是跑还是走的方案数。
f[i][1]=f[i-k][0],f[i][0]=f[i-1][0]+f[i-1][1]

答案即为$\sum_{i=l}^{r} f[i][0] + f[i][1]$，记录前缀和即可。

B discount

题目大意

有n种饮料，第i种饮料的价格是每瓶p[i]。
购买一瓶第i种饮料时，你可以在以下两种优惠中选择一种：
1.该饮料优惠d[i]元
2.免费送一瓶第f[i]中饮料
问最少花多少钱使得每种饮料都买或送一瓶。

做法

饮料之间的赠送关系形成了一棵基环内向树结构。
先考虑树的问题怎么做？

Dp[i][0]表示i的子树内每种饮料都至少一瓶的最小花费。
DP[i][1]表示i的子树内每种饮料都至少一瓶的最小花费，并且以第二种优惠方式购买了第i种饮料

那么就可以得到转移

$$DP[i][1] = p[i] + \sum_{t \in son[i]} dp[t][0]$$

$$Dp[i][1] = \min\left(p[i] - d[i] + \sum_{t \in son[i]} dp[t][0], min_{x \in son[i]}\left(dp[x][1] + \sum_{t \in son[i], t!=x} dp[t][0]\right)\right)$$

B discount

做法

如何处理环的问题呢？
枚举环上任意一点的状态，然后断环为链做一个DP。

假设环上的点是$circle_{1..m}$，先把环上每个点挂出去的子树的DP值求出来，那么在做链的DP时就有以下转移式

$$G[circle_i][1] = G[circle_{i-1}][0] + dp[circle_i][1]$$

$$G[circle_i][0] = \min(G[circle_{i-1}][1] + \sum_{t \in son[circle_i]} dp[t][0], G[circle_{i-1}][0] + dp[circle_i][0])$$

如果环上第一个点不是靠环上的最后一个点附赠而来，则$G[circle_1] = dp[circle_1]$
然后取$G[circle_m][0]$更新答案
如果环上的第一个点是靠环上最后一个点附赠而来，则$G[circle_1][0] = \sum_{t \in son[circle_1]} dp[t][0]$
然后取$G[circle_m][1]$更新答案

C message

题目大意

有一个无线大的平面，平面上有n条斜率不为零的直线。
有m次询问，每次询问从y轴上的一个点出发往x轴正方向沿一条直线走，最后一个与这n条直线相交的点的横坐标。
n,m<=50000

做法

一条直线的表达式为ax+b，一次询问线的表达式为cx+d，则这两条线的交点是$-\frac{b-d}{a-c}$。
如果我们把它们看做两个点(a,b)(c,d)，则直线交点就是现在这两个点的斜率的相反数。

问题就转化成了：平面上有若干个点，每次给一个点，问这个点到平面上所有点的斜率最小值。

这个就可以用凸包进行维护了。

D money

题目大意

你要按照顺序依次经过n个商店，每到达一个商店你可以购买一件商品，也可以出售你手中的商品。
同一时刻你手上最多拿一件商品。在第i个商店购买和出售的代价都是a[i]。
问你经过完n个商店后的最大收益。
同时，在最大化收益的前提下，求最小的交易次数。

做法

DP

f[i][0/1]表示已经访问完了i个商店，你手中是否有商品，此时的最大收益。
g[i][0/1]表示当f[i][j]取最大值时最少的交易次数。

贪心

首先，如果a[i]=a[i+1]，则可以删掉第i+1个商店。因为任何在第i+1个商店进行的交易都可以转为在第i个商店进行，且收益不变。之后，如果a[i]<a[i+1]，则离开第i个商店时一定要带上一件商品。如果a[i]>a[i+1]，则离开第i个商店时一定要空着手。这样，第一问的答案就为$\sum_{i=1}^{n-1}\max(a[i+1]-a[i],0)$，第二问的答案就为长度>1的极大递增连续段的数量。

E tree

题目大意

一棵n个结点的树，每个点有一个点权，有m次操作，每次操作有三种：1.修改一个点的点权 2.修改一个点的父亲 3.询问包含某个点的所有大小为c的连通块的点权乘积之和。
n,m<=100000,c<=10

做法

一组询问怎么做？

DP

f[i][j]表示以i为根的子树中，所有包含点i的大小为j的连通块的点权乘积之和
如何转移？
DP时如果b为a的孩子，则把子树b的信息合并到子树a中

```
for (int i=10;i;i--)
        for (int j=1;j<i;j++)
                f[a][i]=(f[a][i]+1LL*f[a][j]*f[b][i-j])%mod;
```

注意到，这个DP是可逆的
DP时如果b为a的孩子，且子树b的信息已经被统计进入a中，我们可以通过逆DP把子树b的信息去掉

```
for (int i=1;i<M;i++)
        for (int j=1;j<i;j++)
                f[a][i]=(f[a][i]-1LL*f[a][j]*f[b][i-j])%mod;
```

E tree

发现了这个性质后，我们就可以解决修改和多组询问的问题了。

对于修改点权和修改父亲操作。因为c<=10，所以一个点的权值最多只会影响到它的10级以内的祖先。
假设这些祖先从下到上分别为 fa[1..10]，那么我们先对于每个i，把dp[fa[i+1]]的信息去掉dp[fa[i]]的部分。
然后就进行改点权或者换父亲操作，做完之后，再依次把dp[fa[i]]的信息合并到dp[fa[i+1]]中。

对于询问操作，我们可以进行一个换根的DP。本来DP的根是a，如果我们想切换到a的一个孩子b，则
首先把f[a]中的信息去掉f[b]的部分，再把f[b]中的信息加上f[a]的部分就可以了。

复杂度n*10^3

F trade

题目大意

有n个仓库，第i个仓库初始有a[i]单位的货物。
依次有m个订单，白兔会开一辆初始为空的车依次经过s[i]个仓库x[i][0..s[i]-1]。到达一个仓库时可以把车上的货物放到仓库中，也可以把仓库中的货物放到车上。访问完这些仓库后，车中剩下的货物就会被送到顾客家。
第i个订单的顾客有一个需求上限lim[i]，表示最多卖lim[i]件货物给这个顾客。
问在这m个订单中能出售的货物量总和的最大值。
同时有k个干扰器，每个干扰器的作用范围是一个圆。对于一个订单经过的某一个仓库，如果它和顾客家的连线与圆相交，则车会跳过这个仓库。
n,m<=1000,k<=10

做法

首先，通过简单的几何计算，我们可以求出哪些仓库会被skip掉，接下来就转化为了没有干扰器的问题。

接下来如何做呢？
按照订单的顺序一个一个扫描，对于每个仓库i，我们维护一个布尔数组e[i][1..n]，e[i][j]表示之前能否通过车辆的运输把第j个仓库的货物运送到第i个仓库。
对于当前订单经过的点x[0..s-1]，之前如果第a个仓库的货物能够运送到某个x[b]，则可以在这一轮中运送到x[b+1],x[b+2]..x[b+s-1]，这样，就相当于对于每一个t，e[x[b+1]][t]|=e[x[b]][t]。
这样我们就实时维护了一个仓库的货物能否运送到另一个仓库，对于第i个订单，只要第z个仓库的货物能运送到x[s-1]仓库，那么就可以卖给第i个订单的顾客。

F trade

题目大意

有n个仓库，第i个仓库初始有a[i]单位的货物。
依次有m个订单，白兔会开一辆初始为空的车依次经过s[i]个仓库x[i][0..s[i]-1]。到达一个仓库时可以把车上的货物放到仓库中，也可以把仓库中的货物放到车上。访问完这些仓库后，车中剩下的货物就会被送到顾客家。
第i个订单的顾客有一个需求上限lim[i]，表示最多卖lim[i]件货物给这个顾客。
问在这m个订单中能出售的货物量总和的最大值。
同时有k个干扰器，每个干扰器的作用范围是一个圆。对于一个订单经过的某一个仓库，如果它和顾客家的连线与圆相交，则车会跳过这个仓库。
n,m<=1000,k<=10

做法

接下来就变成了一个网络流模型：
一个二分图，左边建n个点表示n个仓库，S向这n个点连a[i]。
右边建m个点表示m个订单，这n各点向T连lim[i]。
如果第i个仓库的货物能够运输给地j个订单的顾客，则左边第i个点向右边第j个点连边oo。
最大流即为答案。
之前e的操作可以使用bitset优化，那么复杂度就是
n*m*n/32+maxflow(n+m,n*m)。

IG transform

题目大意

数轴上有n个集装箱，第i个集装箱位于坐标x[i]，有a[i]件货物。
现在要把集装箱进行一些移动，求在所有货物移动总距离不超过T的情况下，最多能把多少个集装箱移动到同一个位置。

做法

因为我们要让货物移动总距离尽可能小，所以最后所使用的集装箱的初始位置在数轴上一定是一段区间。
如果固定了这个区间，那么最优方案就是把这些集装箱移动到这些集装箱的坐标中位数的位置。

答案满足可二分性，先二分答案。然后我们按照从左至右的顺序枚举区间的左端点，那么区间的右端点和区间的中位数都是单调递增的，一遍枚举一遍维护即可。
复杂度O(n*log(sum(a[i])))

H travel

题目大意

一棵树，点有点权。要选出三条不相交的链使得三条链的权值之和最大。

做法

使用树形DP
f[i][j]表示以i为根的子树选了j条路径的最大权值和
g[i][j]表示以i为根的子树选了j条路径加一条包含i的竖直链的最大权值和。

H travel

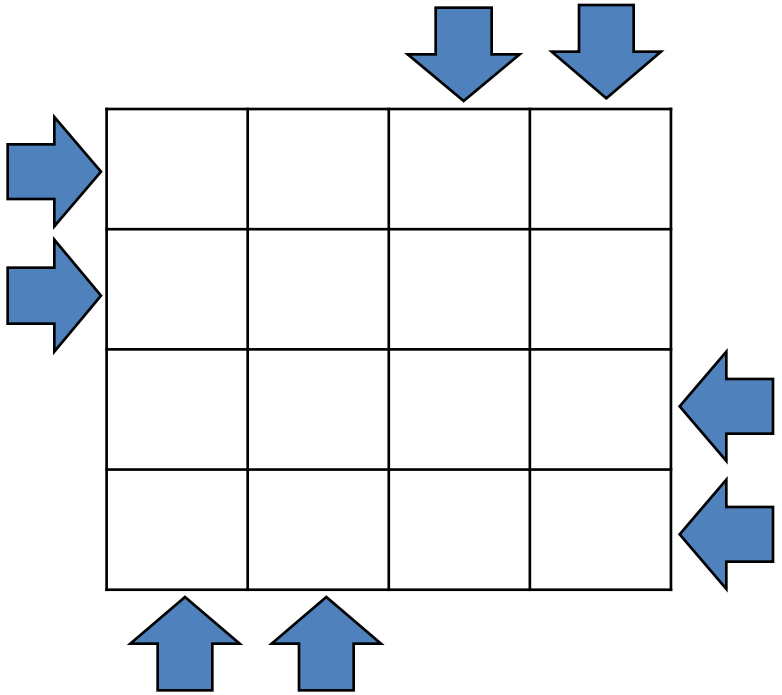
```
int n, w[N];
ll f[N][4], g[N][4];
vector <int> e[N];

void dfs (int u, int fa = 0) {
        ll a[4][3] = {0}, b[4][3] = {0}, c[4][3] = {0};
        int x, y, p, q;
        for (int v : e[u]) if (v != fa) {
                dfs(v, u);
                memset (b, 0, sizeof(b));
                for (x = 0; x <= 3; x ++) b[x][0] = f[v][x], b[x][1] = g[v][x];
                memset (c, 0, sizeof(c));
                for (x = 0; x <= 3; x ++) for (y = 0; x + y <= 3; y ++)
                        for (p = 0; p <= 2; p ++) for (q = 0; p + q <= 2; q ++)
                                chkmax(c[x + y][p + q], a[x][p] + b[y][q]);
                memcpy (a, c, sizeof(c));
        }
        for (x = 0; x <= 3; x ++) chkmax(f[u][x], a[x][0]);
        for (x = 1; x <= 3; x ++) for (y = 0; y <= 2; y ++) chkmax(f[u][x], a[x - 1][y] + w[u]);
        for (x = 0; x <= 3; x ++) for (y = 0; y <= 1; y ++) chkmax(g[u][x], a[x][y] + w[u]);
}
```

使用树形DP
f[i][j]表示以i为根的子树选了j条路径的最大权值和
g[i][j]表示以i为根的子树选了j条路径加一条包含i的竖直链的最大权值和。

I car

题目大意



你要在一个n*n的矩形的边界上方若干辆车，所有车从同一时刻出发，以同样的速度，从某一列的一侧开到另一侧或者从某一行的一侧开到另一侧。问最多放多少量车使得存在一种方式，这些车在行驶的过程中互不相撞。
（车可以视为质点）
同时还会有若干个格子被损坏车辆不能开进被损坏的格子。

做法

先考虑所有格子全部完好的情况。
通过爆搜/脑洞，发现答案就是2n-(n mod 2)。

如何证明？
首先，一行一列最多只能放一辆车，同时如果n为奇数，则第(n+1)/2行和第(n+1)/2列不能都放车，所以刚刚那个值是答案的上界。
同时我们也可以很容易地构造一组满足这个解的方案，得证。

有格子损坏时的情况也不难处理，就把那些行列去掉就好了。

J farm

题目大意

有一个n*m的矩形，每个位置有一个数。有T次操作，每次往一个子矩形的每个格子中放入一个数。
求有多少个格子中被放入了至少一个与对应位置不相同的数。
n*m<=1e6,T<=1e6

做法

先考虑一个特殊的情况：矩形中的数和T次操作放的数都为0或1。
对于这种情况，我们只需要用矩阵前缀和统计一下每个格子被多少个0覆盖，被多少个1覆盖。
如果一个格子的数为0且被放入了至少一个1或这个格子的数位1且被放入了至少一个0则就会对答案产生贡献。

然后考虑原问题。
如果某个格子的数是i，而它被放入了至少一个j，且i!=j，则需要统计进入答案。
注意到，i!=j则i和j至少有一个二进制位不相同。
我们枚举0~19的每一个二进制位，然后把所有数字按照这一位是0还是1划分成两个集合，就变成了上述特殊情况的问题。一个格子只要至少在某一个二进制位的子问题时被统计进入答案，就加到总答案中去。
复杂度 (nm+T)log(nm)

J farm

另一种做法：

我们给每种植物花费随机一个[1e6,1e7]内的权值，然后对于每种植物，求覆盖它的所有化肥的权值之和。
如果这个值是这个格子权值的倍数，则有很大概率只有同种化肥覆盖了当前格子，否则没有。

然后我们可以尽量只随机质数(所以权值只弄到1e7内，可以预处理所有质数)。
这样，如果一个格子权值是p，那么正确率就是$\frac{p-1}{p}$。如果多做几次就几乎没有问题了。

K carpet

题目大意

有一个n*m的矩阵A，每个位置有一个字符和一个权值，现在要找一个子矩阵，使得这个子矩阵是A的一个循环节，并最小化子矩阵的权值最大值。

做法

首先要找到矩阵的一个最小的循环节，假设是p和q，那么最优解就一定是选一个p*q的子矩形。

如何求最小的循环节？行和列可以独立考虑。
求列的循环节时，我们可以对每一行做一次kmp，找到这些行的最大公共循环节。
求行的循环节时，我们可以对每一列做一次kmp，找到这些列的最大公共循环节。

接下来问题就变成了，求所有大小为p*q的子矩形的最大值的最小值。可以采用单调队列，求出所有p*q的子矩形的最大值，然后找一个最小的即可。

Thanks