

A 雷神之路

题意

在一个坐标轴上面起始位置是0，你可以往右走一步，两步，三步。其中某些位置不能走，问你走到位置n有多少种方法（ $1 \leq n \leq 1e18$ ）

分析

- 这题是走楼梯的进阶版，状态很好想 $dp[n]$ ，转移有三种：走一步两步三步转移过来。不过由于n太大，很容易想到矩阵加速。
- 用矩阵A表示第i个可走，矩阵B表示第i个不可走。有些地方x不能走，那么我只要用矩阵加速求出第x-1个矩阵C，那么第x个矩阵就是 $C*B$ ，最后得出第n个矩阵

思考

- $dp[n] = dp[n-1] + dp[n-2] + dp[n-3]$ ($n \geq 3$) 所以我们第一个矩阵是从2开始的，小于2的方案要手动打出来

B Snowd 修长廊

题意

- 现在要修长廊覆盖n个点，第i个点可以单独修一个长廊，也可以修到前面第j个点（覆盖j到i之间的点），现在修i和j之间的长廊花费 $cost(i,j) = W + (x_i - x_j)^2$ ，其中W是固有花费，现在求最小花费是多少

分析

- 我们状态很好建立， $dp[i]$ 表示覆盖到第i个点为止至少需要花费多少，但是他可以有很多状态转移过来。

```
dp[i] = min(dp[j-1]+cost(j,i));
```

- 如果现在有两个状态j和k(其中 $j > k$)可以转移过来.那么用单调队列dp思想把这两个相减

```
dp[j] + cost(j,i) - dp[k] - cost(k,i)
=(dp[j] + x[j]*x[j])-(dp[k] + x[k]*x[k]) - 2*x[i]*(x[j]-x[k])
```

那么也就是说如果 $dp[j]$ 转移过来比 $dp[k]$ 更优的话,就有上式小于0,那么我们在更新最优的 $dp[y]$ 的时候,如果前面存在一个状态x比状态i差,那么便有上式成立

同时可以发现如果现在就已经成立,那么这个式子对y以后的状态都会成立,因为对y以后的xi上式也成立(这个就是斜率dp的特点,总会有一个自变量一直递增或递减),那么状态x就属于无用状态应该从队列中剔除

判定条件一:如果i点没有j点优(也就是说j点比i点更优)的条件是i和j($i < j$)斜率小于 $2x[i]$

- 那么如何保证队列起第一个就是最优的呢?

感性的认识:

根据上面的式子我们可以一直判断第一个和第二个元素然后剔除,从而使得保证队队列第一个一定比第二个更优,那么剩下的就需要保证第一个比第三个,第一个比第四个更优...这个性质一直成立就好了

假设上面这个性质成立则有:23点的斜率大于12点,34点斜率大于12点... 但是确保每相邻两个点斜率大于12点不具有**传递性**,所以我们可以缩小简化:23点的斜率大于12点,34点斜率大于23点...从而形成了**下凸包**

从结果看:斜率递增保证了下凸包,而下凸包其实是保证了:**第三个点没有第二个点优,第四个点没有三个点优..**,因为第一个破坏下凸包性质的点一定不是最优,可参考理性认识的证明

所以我们要维护上面两个性质:1:第一个点比第二个点优,2:第三个点没有第二个点优,第

四个点没有三个点优...(用维护斜率递增实现)**

思考

- 如何证明维护的队列的第一个就是最优的呢?

- 理性认识:

$$dp[i] = dp[j] + x[j]^2 + x[i]^2 + 2x[i]*x[j] + W$$

$$dp[i] - W - x[i]^2 = dp[j] + x[j]^2 + 2x[i]*x[j]$$

从上面这个式子看出 $dp[i]$, $x[i]^2$, W 这三个相对于 i 是定值

问题转换:我们要找到这样一个 $x[j]$ 和 $dp[j] + x[j]^2$ 使得 $dp[i]$ 最小,看成 $b = y - kx$,也就是 b 最小.从而我们可以画出一些点 $(2x[j], dp[j] + x[j]^2)$,用斜率固定 k 的直线从下面往上面开始扫描,第一个接触到的点就是我们要找的最优点.这样找的话就会发现一个特点,上凸包的转折点永远不会被扫到,所以也应该剔除.

如果我们在每一次插入的时候一直这样判断,那么就能够保证最后两个斜率一定大于前两个点的斜率,由于插入斜率大小具有**传递性**,所以能够保证**斜率递增**

但是光斜率递增就一定能够使得第一个最优吗?

确保第一个最优的充分必要条件是每个点和第一个点的斜率大于等于 $2x[i]$,这样看来显然不能够确保第一个最优.(举一个极限的例子,所有点和第一个点的斜率都小于 $2x[i]$ 但是却递增)

为了弥补上面的不足,可以让第一个优于第二个,也就是 12 斜率大于 $2x[i]$,由于上面凸包的性质使得第一个点和所有点斜率都大于 $2x[i]$ 也就是说,以后的点不能够比第一个点更优.所以:**第一个是最优的**

C Taosama和煎饼

题意

- 韬韬想吃煎饼但是有急事,一个煎饼有 n 个工序,每个工序 i 有着 $A[i]$ 美味度,韬韬有 m 个道具,每个道具用一次可以前进 $b[i]$ 个单位,每个工序一旦被跳过得不到美味度,问你使用这些工具最多获得多少美味度($0 \leq b[i] \leq 4, 0 \leq a[i] \leq 100, 0 \leq n \leq 350, 0 \leq m \leq 120$),确保 $b[i]$ 的和是 $n-1$

分析

- 建立状态

$dp[i][a][b][c][d]$ 表示处于 i 位置还剩 $abcd$ 个四种道具(a 表示前进 $b[1]$ 个道具)最多能够获得多少美味度

状态转移

$dp[i][a][b][c][d]$ 可以由 $dp[i-1][a-1][b][c][d]$ 等四种状态转移过来

- 状态优化

上面的状态很容易想,但是 $350 * 40 * 40 * 40 * 40$ 复杂度很高,我们可以发现,这五个变量只要确定四个,那么第五个变量是确定的,所以我们可以减少一维, $dp[a][b][c][d]$ 就好了,复杂度 $40 * 40 * 40 * 40$ 完美过题

- 超时原因

由于建立状态思维定式,所以建立的状态是 $dp[350][40][40][40]$,超时了,但仔细发现这里的 350 有很多状态达不到

思考

- 技巧

- 对于这种固定形式的转移,记忆化搜索写起来最快
- 建立状态的时候如果有多种选择首先找范围小的,其次找容易实现的

D 任务

题意

- 有两台机器n个任务,每个任务i在机器A完成时间是a[i],B上面完成时间是b[i],任务i可以被处理当前仅当每个任务j(i>j)已经被完成或者正在进行.求最少完成任务的时间(n<2000,a[i],b[i]<=3000)

分析

- 首先读清楚题目意思

任务i可以被处理当前仅当每个任务j(i>j)已经被完成或者正在进行

上面这句话有什么用?

仔细分析后可以发现上面这个性质确保了执行任意任务的时候机器AB时间差不会超过3000

假如A比B将少运行时间超过3000,那么现在放置任务i+1

- 那么任务i+1不能放置机器A最开始空闲的时候,因为这样B机器必然(B多A3000时间)有任务j(j<i)还没有没执行;那么任务i+1只能当道B机器运行最后一个任务j的起始时间上去,这样的方案必然没有比任务j放置到A机器方案优(可以画图模拟一下)
- 把任务i+1放置到机器B,那么这种方案将继续造成恶性循环又进入此状态,显然最终这种方案不优

所以任一个任务选择的时候AB机器运行结束时间差不能够3000

建立状态

dp[i][j]表示完成任务i并且机器AB运行结束时间差为j的最少时间,(j>0表示A比B多j,否则B比A多-j,后面右移3000,保证都是正数)

状态转移

dp[i][j]由dp[i-1][j+a[i]]和dp[i-1][j-b[i]]转移过来,其中要注意任务i放到机器A或B中可能对时间没有影响

思考

- 错误分析
 - 没有对非法状态进行限制,比如状态转移的时候,放置任务i到机器A,必须设定任务i时间当值 >= B机器最后一个任务j放置时间!!!确保i之前的任务被处理完或正被处理
 - 没有考虑任务i放置到机器A后对时间无影响的情况

E goozy搭积木

题意

- goozy对积木十分的狂热,今天他想搭一个双子塔(就是两个高度一样的塔)!他想知道,用现有的积木,能不能实现这个想法.积木个数n(1 <= n <= 50),每个积木的高度hi(1 <= hi <= 500000),题目保证所有积木高度总和不超过500000。

分析

- 建立状态

这题和能不能把所有的平分是不一样的,因而我只要保证有一种方案把积木部分平分就好了

所以建立状态不能用dp[i][j]表示i木块j高度是否可达

状态是dp[i][j]表示前i个木块使得两份木块高度差为j的最大可达高度是多少。

- 状态转移

状态转移就很简单了,积木i放到比较高的一方或者比较低的一方

放到比较高的一方

```
dp[now][j+h] = max(dp[now][j+h], dp[1-now][j]+h);
```

放到比较低的一方

```
newh = max(dp[1-now][j]-j+h, dp[1-now][j]);
```

```
dp[now][abs(h-j)] = max(dp[now][abs(h-j)], newh);
```

思考

- 有时候dp建立状态后发现某一位范围过大,不能直接dp[i][j]的时候可以考虑特殊性质,然后对他们的差进行状态转移,正如D题一样

F 先锋看烟花

题意

- 一共有n个地点（1到n排列），一共有m个烟花，每个烟花放出的地点ai和时间ti，每个烟花有观赏值bi，对于每个烟花，对先锋的幸福度贡献bi-|ai-cur|，其中cur表示放第i个烟花时先锋所处的位置，因此，当先锋离烟花太远时，幸福度甚至会下降！已知先锋每秒最多运动d距离，问这个晚上先锋看烟花能获得的最大幸福度。

分析

- 这相当于是一道移动接苹果的题目,只不过每秒移动范围为d并且烟花放的每一个点都有一个贡献值可为负数

- 状态建立

dp[i][j]表示第i个烟花位于j位置最多能够获得多少幸福感

状态转移

```
dp[i][j]=min(dp[i-1][k]+bi-|ai-j|);
```

k和j差值不超过(t[i] - t[i-1]) * d

转移优化

直接转移显然不行,那么就可以分成两种情况并进行单调队列优化了 若ai>cur

```
dp[i][j]=min(dp[i-1][k]+bi-ai+j);
```

那么维护一个和j差范围处于(t[i]-t[i-1])*d的队列

如果 dp[i-1][k] 和 dp[i-1][kk] 都在队列中并且 $k < k \&\& dp[i-1][k] < dp[i-1][kk]$ 那么在对i时刻烟花j以后的位置,都不会选择dp[i-1][k]了,因为dp[i-1][kk]一定比他优,所以dp[i-1][k]这个状态就要从队列中间pop出来

所以队首的只要处于范围之内一定最优

这个队列最多插入n次pop n次,因而复杂度是O(n)

思考

- 当遇到一种可以由很多种状态转移过来的时候,考虑一下nlogn线段树或树状数组查询或单调队列或斜率优化

H 又见背包

题意

- 有n种大小不同的数字a_i,每种m_i个,判断是否可以从这些数字中选出若干使它们的和恰好为k。(0<n<=100,0<k<=1e5)

分析

- 建立状态

dp[i][j]表示前i个物品能不能组成和是j

- 状态转移

$dp[i][j] = dp[i-1][j-a[i]];$

- 状态优化

上面的状态首先可以二进制对个数进行优化,然后可以发现单独的一次转移居然只是转移一个bool,太浪费时间了,所以充分使用int或ll的位数,我们可以把j分成64位一组,一次性转移一组这样转移复杂度就除以64了相当于降了两个数量级了

现在复杂度是 $nk\log m/64$,这样相当于消去 $\log m$ 而且转移过程还是位运算很快,所以就过了

思考

- 如果以后遇到bool为dp存的状态的时候,那么这个状态肯定是不好的,这时候要么像这题一样压缩一下转移,要么重新建一个状态

I Mingo的游戏

题意

- Mingo 最近迷上了一款电子游戏。这款游戏有N个关卡，编号为1到n。这些关卡可以被玩家分为K($1 \leq K \leq N$)组，每组包含至少一个并且编号连续的关卡。这款游戏有些很奇怪的规则：

1.如果所有的关卡都被通关，那么这个游戏立即结束，否则系统会找到第一个包含还没有通过的关卡的那组，设X是这组的编号。

2.设在X组里，已经通关的关卡编号为 $i, i+1, i+2, \dots, j$ 。那么 $j+1$ 为最早的一个没有通关的关卡。每一个关卡都有一个权重 t_i 。此时系统会随机以 t_i 的概率进入关卡k。

3.Mingo是一个富有天赋的玩家，每一关都能轻松过关，但是每一关都会消耗她一个小时去过关。

Mingo想以最快的速度通关所有的关卡，但是她非常困惑到底如何分组才能使她尽可能的尽早通关所有关卡呢。你能帮她计算一下如何分组才能使她通关所有关卡所用时间的期望最小么。

分析

CF原题,第一次遇到的斜率dp,详解见B

思考

- 这题后面我CF过了,但是scu一直过不了,可能因为我写错了,然后CF用相对误差,SCU用绝对误差

L 来签个到吧

题意

- 给你 n ($2 \leq n \leq 60,000$) 个球,每球上都写有互不相同的数字 t ($0 \leq t < 100,000$),这些球放在一个盒子里.开始你能执行一种加球操作:选择任意两个球: x 和 y ,然后看 $|x - y|$ 在已有的球中是否存在,如果不存在,就把 $|x - y|$ 写在一个球上,把这个球加入这个盒子,这里就成功完成了一次加球操作. 你就一直加球,直到盒子中任意两个数的差,在集合中已经存在

现在摸球,问你摸完所有球的所进行操作次数的期望

分析

- 最终状态会有多少个球？

显然加球操作很像辗转相减的过程,我们只需要对这题求最大公约数,然后最大值除以这个最大公约数就是最终的个数

- 期望怎么求？

如果当前摸过了 i 个球,还剩下 $n-i$ 个就没有摸,那么摸到 $i+1$ 球的期望是多少？

- 可以用无穷级数来求

- 可以感性认识一下, n 个球我要摸到 $n-i$ 球中的一个期望是多少,就是 $n / (n-i)$

所以最后 $ans = \sum(n/i) + \text{加球操作数}$

思考

- 这题要注意对 0 的处理
- 这题实际不难,但是读错题意,以为操作只包括摸球