

[题解]Div2 寒假练习 1

PS标程与题解 提供：

- A~F、U~V 卢婉青
- H~O 马宇昊
- G、P~T 郁鸣

A-Theatre Square-[CF1A](#)

tags: 数学、模拟

简单数学思维题，按照一块地板的边长计算铺设地点的长和宽至少需要几块才能铺完，不够的补上即可。之后将长宽相乘计算结果。注意结果会爆 int 。

时间复杂度: $O(1)$

B-NN and the Optical Illusion-[CF1100C](#)

tags: 数学、几何

题目给出外部圆的数目 n 和内部圆的半径 r ，要求求出外部圆的半径以满足要求。简单数学思维题，推出公式即 $\sin(\frac{\pi}{n}) \cdot (R + r) = R$ 。

需要注意 π 的精度问题。可以使用 `const double PI = acos(-1)`

时间复杂度: $O(1)$

C-Three Friends-[CF1272A](#)

tags: 贪心

若三人在一个点上则不需要移动，否则向中间那个点移动(可以适当推一下他们之间的关系)。

时间复杂度: $O(1)$

D-Snow Walking Robot-[CF1272B](#)

tags: 构造、思维

先遍历统计 LRUD 四个方向的个数

1. 如果只有一个方向 或者 两个对称方向为0，那么意味着移动路线只能是一维的，又因为不能重复走某个点，就值需要走两步即可(例如只有 U 的个数为0，那么只能 LR 或 RL 了)
2. 当然如果出现两个不相对的方向同时为0，则不能移动(如 R 和 U 都为0，则输出0)
3. 如果都不为0，输出路线形成一个矩形就行(两邻边分别为 UD 的数量的最小值和 RL 的最小值)。

时间复杂度: $O(n)$

E-Substrings Sort-[CF988B](#)

tags: 字符串、暴力

字符串问题, 先将一组字符串全部存在字符串组里, 将字符串组排序, 查找前一个字符串是否在后一个字符串能找到, 若都能找到则满足, 否则不满足。

时间复杂度: $O(n^2 \log n)$

F-Blocks-[POJ 2363](#)

tags: 枚举、暴力

令堆积起来后的长宽高为 a, b, c , 则有 $a * b * c = n$, 实质上就是求 $2(a * b + b * c + a * c)$ 的最小值, 也就是 $2(n/a + n/b + n/c)$ 的最小值。显然当 a, b, c 最接近的时候面积最小, 故可以从小到大枚举 a, b, c 这样最后一组可行解一定为最优解。注意 $n=0$ 的情况需要特判。

时间复杂度: $O(n)$

G-Best Cow Fences-[POJ 2018](#)

tags: 二分、前缀和

若长度为 L 的子段中每个数减去平均值, 便可以转换问题为“求一个长度不小于 L 的子段和最大”。所有用二分的形式枚举平均值。因都是正整数, 子段和最大必定 $sum(l, r) \geq 0$ 。

检查长度大于等于 L 的子段, 可以前缀和相减的形式。设 A_i 为 $a[i] - mid$ 的结果, sum_i 表示 $A_1 + \dots + A_i$ 的和, 即:

$max_{i-j \geq L} (A_{j+1} + A_{j+2} + \dots + A_i) = max_{L \leq i \leq n} (sum_i - min_{0 \leq j \leq i-L} (sum_j))$ 。公式中, 随着

i 的增长, j 的取值 $[0, i-L]$ 也会增长。所以 $min_{0 \leq j \leq i-L} (sum_j)$ 只需要每次记录的最小值 $curmin$ 和新加入的 $sum[i-L]$ 比较取最小即可。

check函数代码如下:

```
bool check(double mid) {
    for (int i = 1; i <= n; i++) {
        A[i] = a[i] - mid;
        sum[i] = sum[i - 1] + A[i];
    }
    double curmin = INF, ans = -INF;
    for (int i = L; i <= n; i++) {
        curmin = min(curmin, sum[i - L]);
        ans = max(ans, sum[i] - curmin);
    }
    if (ans >= 0) return true;
    return false
}
```

本题二分时要注意二分上下界和精度的取值。EPS建议在 10^{-4} 以上。

时间复杂度：check函数 $O(n)$ ，总体复杂度 $O(n\log n)$

H-Color the ball-[HDU - 1556](#)

tags：前缀和、差分

本题考虑有暴力更新的复杂度： $O(1e5 * 1e5)$ ，对于3s的时间来时显然不够的，于是可以引入差分数组。

差分数组的运用就是对前缀和的一种运用，

举个例子：先设定一个数组 a

9 1 8 3 5

第一步：将每一个数字减去前一个数字得到第一个数组 b

9 -8 7 -5 2 (9是第一个数字，所以可以将其前一个数字看为0, 9-0=0)

前缀和在此处的用法就是得到元原来的数组a

$$a_1 = b_1 = 9$$

$$a_2 = b_1 + b_2 = 9 + (-8) = 1$$

$$a_3 = b_1 + b_2 + b_3 = 9 + (-8) + 7 = 8$$

.....

如果我们要让 b2, b3 同时加上 1

那么，我们只用将 b2 加上 1 就好

$$b_2 = b_2 + 1$$

$$b_1 + (b_2 + 1) = 9 + (-8 + 1 = -7) = 2 = a_2 + 1$$

$$b_1 + (b_2 + 1) + b_3 = 9 + (-8 + 1 = -7) + 7 = 9 = a_3 + 1$$

但是我们可以看到在 b2 加上 1 后，除了 a2 和 a3 会加1以外后面 a4, a5 也会加上 1 为了抵消这种影响所以，我们同时要给 b4 减去 1

比如

$$b_4 = b_4 - 1$$

$$b_1 + (b_2 + 1) + b_3 + (b_4 - 1) = 9 + (-8 + 1 = -7) + 7 + (-5 - 1 = -6) = a_4 + 1 - 1 = a_4$$

这道题目中起手每次对一段路进行渲染，也就是意味着每次将一段数字加 1，然后a数组在这题中就是全为0。

这样可以使得区间更新的 $O(n)$ 变成 $O(1)$ 的 单点更新。

时间复杂度： $O(n)$

I-小数化分数2-[HDU 1717](#)

tags：gcd、数论、模拟

字符串表示的精确小数转换方法：

数学方法，以10的对应小数位数次幂为分母，小数部分为分子，然后约分，如0.25转换为 $25/100=1/4$ 。

也就是说对于一个特定的小数只要

0.***** 将小数点后面的数字看作一个相同的整数 ***** (五个数字)

然后除以一个 $\text{pow}(10,5)$ ，100000

然后将其化作 *****/100000，再进行化简即可

化简方法：

$x = \gcd(****, 100000)$ 得到两数的最大公因数

得: $(****/x)/(100000/x)$

特殊方式表示的循环小数转换方法:

单纯的循环小数只要将小数部分除以等位数的9就好

即: $0.(1234) = 1234/9999$

$x = \gcd(1234, 9999)$

得: $(1234/x)/(9999/x)$

组合小数 0.5(6)

将两部分分开计算

1. (将不循环部分提到小数点前面, 方便接下来的计算): $0.5(6) * 10 = 5.(6)$
2. (计算前半部分): $5/10 = 1/2$
3. (计算后半部分): $6/9 = 2/3$
4. (将上面乘的10的影响抵消): $1/2 + (2/(3 * 10)) = 1/2 + 2/30 = 17/30$

同理:

如果向前提位分割时用的是100的话, 下面对影响的抵消也就要用100

本题pow建议自己写一个实现, cmath的 pow 会丢精。

J-寻找素数对-[HDU - 1262](#)

tags: 素数

这是一题很直接的对素数的判断和寻找

首先题目中说了要找最相近的两个素数, 并且这两个素数可以互补成原数

那么:

1. 将这个假设为 n
2. 先找到其中的一个素数, 从 $n/2$ 的地方开始, 因为这是一大一小两个数(将其设做 a 和 b , 且 $a < b$)
所以必然 $a < n/2, b > n/2$, 无论是先找 a 还是先找 b 都可以
3. 一直遍历找到 a , 然后通过 $n-a$ 找到 b
4. 验证 b 也是一个素数

最后得到的 a 和 b 就是答案

(本题的数据范围, 可以直接将所有范围内所有的素数进行标记)

时间复杂度: $O(n)$

K-Yet Another Broken Keyboard-[CF1272C](#)

tags: 组合数学、模拟

这是一道找排列可能性的题目

首先是我们要遍历整个原字符串每次都取连续的一段可用字符串取出来

一: 得到字符串 abacaba, 可用的字符 a b

二: 分割出可用的字符串 $aba|c|aba$, 得到一个无用字符串 c 两个长度位 3 的字符串

三: 排列种类的计算, 3个字符的排列 $= 3 * 2 * 1 = 6$

(n 个字符的排列 $= n * (n - 1) * (n - 2) * \dots * 2 * 1 = n * (n - 1) / 2$)

四: 将所有的得到的种类加在一起 $6 + 6 = 12$

时间复杂度: $O(n)$

L-最少拦截系统-[HDU - 1257](#)

tags: 贪心、DP

有两种方法, 贪心和DP。

方法一 贪心

这道题要得到最少所需要的拦截系统, 这是一种比较简单的贪心思路
我们以题目中的例子为例: 8-389 207 155 300 299 170 158 65

一: 这是我们可以从这个例子中找到的最简单的一种可能性

拦截系统1:389 207 155

拦截系统2:300 299 170 158 65

二: 上面的这个例子虽然可行, 但是从贪心的角度上讲并不是最优的, 所以我们可以得到更优的解
比如说:

最后一发65高度的导弹, 其实可以不用系统2进行拦截, 可以用系统1进行拦截

因为两个系统之前的最低高度为155和158

如果选择了155的系统进行拦截可以在不增加新的系统的情况下, 令下一次的拦截高度更高

拦截系统1:389 207 155 65

拦截系统2:300 299 170 158

三: 举一个简单的例子, 现在有一个新的导弹, 其高度为157

如果我们用前一种拦截方式, 那么这么导弹就无法直接拦截, 需要增加新的拦截系统

拦截系统1:389 207 155

拦截系统2:300 299 170 158 65

拦截系统3:157

而用后一种拦截方式, 可以直接用系统2进行拦截

拦截系统1:389 207 155 65

拦截系统2:300 299 170 158 157

方法二 DP(LIS)

经典LIS(最长上升子序列)问题。有朴素的 n^2 和二分 $n\log n$ 的方法。这里主要讲解 n^2 复杂度的思路。二分的方法, 大家有兴趣可以自己尝试下。

设 $dp[i]$ 表示以 $a[i]$ 结尾的最长上升子序列的长度, 将 $dp[i]$ 初始化为 $dp[i]=1$, 因为 $dp[i]$ 至少为1。

分为两种情况:

1. 是只包含 $a[i]$, 则此时 $dp[i]=1$;
2. 是在 $j < i$ 且 $a[j] < a[i]$ 时, 有 $dp[i]=dp[j]+1$;

所以该题的状态转移方程可大概表示为 $dp[i] = \max(1, dp[j] + 1, j < i \text{ 且 } a[j] < a[i])$

M-Humble Numbers-[HDU - 1058](#)

tags: 搜索、贪心

这题寻找因子组只有2,3,5,7的数字，从举例中我们可以看出第5842个符合条件的数字为2000000000，所以暴力的方法一定会超时
然后我们就可以用循环距离的方式
有两种办法，一种是对组成的因子个数进行循环，每次循环4个数字，另一种是对2,3,5,7进行循环，每次循环到底

1. 2000000000约等于2的30次左右
2. 我们对两种循环方式进行分析
 - 对因子的个数进行循环（从1->30，重复4次，从1个2到30个2，依次类推），所需最大次数30的4次方
 - 对2,3,5,7进行循环（每次遍历2,3,5,7四个数字，遍历30次），所需的最大次数4的30次方
3. 很明显4的30次方要远大于 30^4 ，所以我们选择第一种循环方式
同时可以用 **优先队列** 进行排序，全部记录后再按序输出

时间复杂度： $O(n \log n)$

N-Oil Deposits-[HDU - 1241](#)

tags: dfs、连通块

DFS找连通块的裸题。

这题我们要找的每一块油田就相当于一个整体，每一次找到新的油田就进行统计和标记

1. 首先我们要存图，按照自己喜欢的方式，怎么简单怎么来，设油田的总数初始值为 0
2. 我们要遍历存下来的整个图，每当遇到字符 @ 就代表着找到了一个新的油田，首先总数 sum 上加 1 然后将这一部分的油田进行标记，防止后续遇到同一个油田的时候反复计算
3. 注意这个油田的连接方向是八个方向的

时间复杂度： $O(n * m)$

O-Red and Black-[HDU - 1312](#)

tags: dfs

一道简单的走迷宫的题目，再走迷宫的同时要统计能走的迷宫的路

1. 存好图，黑色瓷砖等于通路，红色瓷砖等于墙壁，字符@为初始位置
2. 用递归的方式从初始位置向各个方向延伸，因为没有说不能重复走已经走过的路，所以只要是连通的路全都可以走到
3. 同上一道题每到新的瓷砖上就进行统计和标记，防止重复计算

P-[N皇后问题](#)-[HDU 2553](#)

tags: dfs、剪枝、打表

$n * n$ 的棋盘下 n 个棋，则第一个棋子一定下在第一行，所以遍历第一行 n 种情况，往下符合题意地蔓延。可以用一个一维数组来保存每行的皇后所在的列，降低空间复杂度。

注意这道题需要提前打表，否则每询问一遍就要重新开始索素，外部数据很大，提交会超时。打表记录下 $dfs(1..2..10)$ 的所有结果，到询问的时候直接 $O(1)$ 输出结果即可。

时间复杂度：加了部分剪枝，计算有点复杂就鸽了。。

Q-Dungeon Master-[POJ 2251](#)

tags: 三维迷宫问题、bfs

*BFS*模板题。从二维迷宫扩展到了三维，只需要增加前后两个 z 轴方向，注意边界和起点终点就可。小坑点，本题数组开到 $30 * 30 * 30$ 很有可能会*Runtime Error*。空间足够，多开点就行了。

时间复杂度： $O(L * R * C)$

R-A计划-[HDU 2102](#)

tags: 三维迷宫问题、bfs

同上题一样，三维的*BFS*。可以设地图为 $mp[2][n][m]$ ，本题对于层次的转换可以使用异或 \wedge 运算方便进行。如： $z \wedge = 1$ 。

题目有个小坑点：

1. 一个传送门对面有可能也是一个传送门。
2. 传送门对面如果是墙则会撞死。
3. 给定的时间 T 可能为负数，在搜索前要经行判断。
4. 时间的限制，需要部分剪枝，若当前点的时间已经大于 T ，则该点不经行搜索扩展。

对于1, 2的情况，我们可以使用数组记录访问过的点，对于访问过，或者本身是墙的点，不加入队列经行扩展。

时间复杂度： $O(N * M * 2)$

S-Prime Path-[POJ 3126](#)

tags: bfs、素数删

*BFS*模板变形。先用素数删求出 $[1000, 9999]$ 以内的素数。对于一个数 x 分别枚举它的4位数字，更改其中一位，可以组成新数字 y ，比对这个数是不是素数，如果是就将它加入到队列中，并记录该数字是第几次。循环里每次处理的都是当前数字 x ，加入队列的是新数字 y ，这样一层一层的扩展，这才是广搜的核心。

时间复杂度： $O(n * 4 * 9)$

T-Find a way-[HDU 2612](#)

tags: bfs

求Y和M到同一家KFC的最短距离之和。稍微烦一点的*BFS*模板题。

分别从Y点和M点开始。进行两次BFS，求出Y、M到各个@的距离记录为 $dis1[i][j]$ 和 $dis2[i][j]$ ，然后求出遍历求出同时可达点中，距离之和最小的。

时间复杂度： $O(n * m)$

U-机器人走方格-[51Nod 1118](#)

tags：DP、递推

递推类DP问题，机器人只能从左上到右下只能向右或者向下走，说明一个位置只能从上或左走来。设 $dp[i][j]$ 为从左上角到 (i, j) 的方案数。那么我们从左上角开始dp， $dp[i][j]$ 等于 $\max(\text{左边的dp值加上边的dp值, 原本点的dp值})$ 。

即状态转移方程为： $dp[i][j] = \max(dp[i][j], dp[i-1][j] + dp[i][j-1])$

注意取模问题。

时间复杂度： $O(n * m)$

V-Bone Collector-[HDU 2602](#)

tags：01背包、dp

01背包问题裸题。用动态规划的思路，阶段就是“物品的件数”，状态就是“背包剩下的容量”，那么很显然 $f[i, v]$ 就设为从前 i 件物品中选择放入容量为 v 的背包最大的价值。

状态转移方程为： $f[v] = \max(f[v], f[v - w[i]] + val[i])$

考虑到用 $f[v]$ 来保存每层递推的值，只要我们在求 $f[v]$ 时不覆盖 $f[v - w[i]]$ ，那么就可以不断递推至所求答案。

时间复杂度： $O(VN)$