NVCPC 2024

解题报告

LZY

SYSU

2024 年 4 月 27 日

目录

1 A. 时间复杂度

2 B. 寻觅

3 C. 声望

4 D. 区区整除

5 E. 采糖果

6 F. 绝对霸主

8 H. 罗德岛战役

9 I. 第三心脏

10 J. 回忆之树

11 K. 骤雨

12 L. 交通建造

13 M. 计算

14 N. 不为人知的位运算童谣

A. 时间复杂度

简明题意:

$n \leq 10^5$, 判断五段代码是否会超时

Code A and B

$A : O(n^2) \rightarrow$ TLE

```
int k=0;
for (int i=1;i<n;++i)
    for (int j=i+1;j<=n;++j)
        ++k;
```

$B : O(n\sqrt{n}) \rightarrow$ AC

```
int k=0;
for (int i=1;i<n;++i)
    for (int j=1;j*j<=n;++j)
        ++k;
```

Code C and D

$$C : O(n \sum_{i=1}^{n} \frac{1}{i}) = O(n \ln n) \rightarrow \text{AC}$$

```
int k=0;
for (int i=1;i<n;++i)
    for (int j=1;i*j<=n;++j)
        ++k;
```

$$D : O(n \log_2 n) \rightarrow \text{AC}$$

```
for (int i=1;i<n;++i) {
    int j=i;
    while (j>0) {
        if (j%2==0) j=j/2;
        else j-=1;
    }
}
```

Code E

$E : O(Fibonacci(n)) \rightarrow$ TLE

```
long long fib(int n) {
    if(n < 0) return 0;
    if(n <= 1) return 1;
    return fib(n - 1) + fib(n - 2);
}
int main() {
    scanf("%d", &n);
    printf("%lld", fib(n));
}
```

$Ans =$ “BCD”

B. 寻觅

简明题意:

求 S 中'sysu' 出现次数，多次单点修改 + 询问

记 $check(i)$ 表示 $S[i...i+3]$ 是否为 “sysu”。
单次修改位置 x，只可能会让 $check(x-3), check(x-2), check(x-1), check(x)$ 发生改变。
对 $S_1, ans = \sum_{i=1}^{n-3} check(i)$ 。
之后每次修改前减去 4 个 $check$ 值，然后修改后再加上新的 $check$ 值即可。
时间复杂度 $O(n+m)$。

C. 声望

简明题意:

已知矩阵 A, 向量 b 。求满足 $Ax = b$ 的 01 向量 x 的个数

直接 dfs 搜索 x 即可。
复杂度 $O(2^n \times m)$

D. 区区整除

简明题意:

正整数 x, y, z, 满足 $x > y > z$ 且任意两个数的**和**被第三个数**整除**。
已知 $x - z$，试构造一组 x, y, z。

结论: $x : y : z = 3 : 2 : 1$。
故当 $d = x - z$ 为奇数无解，否则有唯一解。

证明:

$$\begin{cases} x > y > z > 0 \\ x | y + z \end{cases} \Rightarrow \begin{cases} 2x > y + z > 0 \\ kx = y + z (k \in \mathbb{Z}) \end{cases} \Rightarrow k = 1 \Rightarrow x = y + z$$

$$y | x + z \Rightarrow ty = x + z = y + 2z (t \in \mathbb{Z}) \Rightarrow y = \frac{2}{t - 1} z$$

由于 $y > z$，故 $t = 2, y = 2z, x = y + z = 3z$。

E. 采糖果

简明题意:

已知 $n*m$ 的非负权值矩阵，网格图上设计路径 $P_1=(1,1)\to(n,m)$(只能向右或向上) 和 $P_2=(1,m)\to(n,1)$(只能向左或向上)

求 P_1,P_2 路径并集的权值总和的 max

考虑两人的路径交，必然是一段横着或竖着的线段。预处理出每个点到四个角的最优路线权值。

对于横的线段，精细讨论端点的来源 (保证是两条路径第一次相交)，在一行中设 f_l, g_r 分别为第 l 个元素到左上，左下两条无交路径的权值和，第 r 个元素到右上，右下两条无交路径的权值和，

设 s_i 表示当行权值前缀和，则 $ans = \max_{l \le r} f_l - s_{l-1} + g_r + s_r$，利用单调队列思想，维护 $\max\{f_l - s_{l-1}\}$ 即可。竖的线段同理。

时间复杂度 $O(nm)$ 。

F. 绝对霸主

简明题意:

用 $O(1)$ 空间求序列的绝对众数 (其出现次数 $>$ 序列长度 $/2$)

摩尔投票

将绝对众数映射为 1，其他数映射为 -1，则 $sum > 0$ 。
由此思想得到经典的 “摩尔投票” 算法:
维护前缀和 sum ，以及前缀众数 x 。
初始化 $sum = x = 0$。
当加入新的一个数 y 时:

1. if $sum = 0 : x \leftarrow y$
2. 当 $x = y, sum \leftarrow sum + 1$
   当 $x \neq y, sum \leftarrow sum - 1$

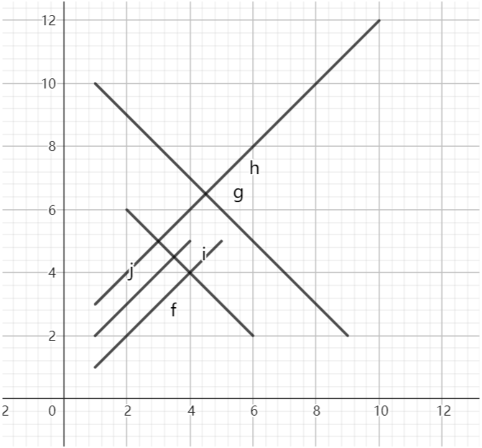
G. 咬合秒针

简明题意:

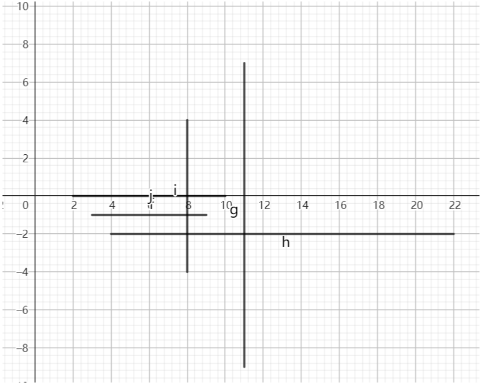
一维相遇问题：已知每个人的移动路线，以及移动起始时间。
所有人的移速相同，求所有**移动中**的人相向相遇总次数。

我们可以做出一个 $x-t$ 图，每个时针视为一个 $(x_i, t_i)->(y_i, t_i+|x_i-y_i|)$ 的线段，在上面表示出每个时针。

如样例 1 ：



现在只需求两种方向线段的交点数量。
我们把整张图旋转 45 度后镜像，原来的每个点 (x, y) 变为 $(x+y, x-y)$，这样就只剩下与坐标轴平行的直线了。



我们可以考虑把所有竖线的 x 位置和横线两个端点的 x 位置从小到大排序，然后从左往右做，遇到左端点时就在对应的 y 位置 $+1$，遇到右端点就 -1，遇到竖线时就统计上那一条直线上所有 y 位置的和，这样就能得出答案了。
维护 y 位置的和需要用能**单点加权，区间求和**的数据结构
(树状数组/线段树/...)

H. 罗德岛战役

简明题意:

对于每个敌人（有物抗、法抗和生命值），每次加入或删去一个特定干员，物理、法术、真实伤害，按照公式计算实际造成伤害，问每次操作后能否击败敌人。

对于每一个干员，按照公式求出它实际造成的总伤，然后用 map,Trie 树，哈希表等数据结构维护编队中的干员及其总伤即可。

I. 第三心脏

简明题意:

给定无向连通图 G，边权最初都为 1，选择若干条无交的迹将边权清零，使得任意两点之间最短路不超过 k

任意在 G 中选出一个生成树 T, 不难发现若 T 上合法, 则 G 上也一定合法.
所以我们只考虑给出连通图的一棵树，如果你了解一个叫做 **树链剖分**的算法，那么可以直接用来解决这一题，因为树链剖分可以将一棵树剖分为若干条重链，根节点到任何节点不会超过 $\log_2 n$ 条轻边，而 $2\lfloor\log_2 2000\rfloor = 20$，实际上不会卡的这么死，可以通过本题。

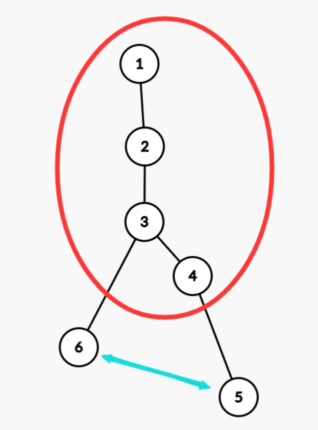
更优的做法：我们对于每个节点 i，设 $f_{i,j,z}$ 表示节点 i 到任意叶子需要经过的未点亮的边数量为 j，当前节点连接所有儿子的边中点亮的边数为 z 条时的一个方案。这样我们可以算出 j 最小的剖分，可以将为点亮的边数缩小到 $\log_3 n$，具体的证明与树链剖分的证明相似。

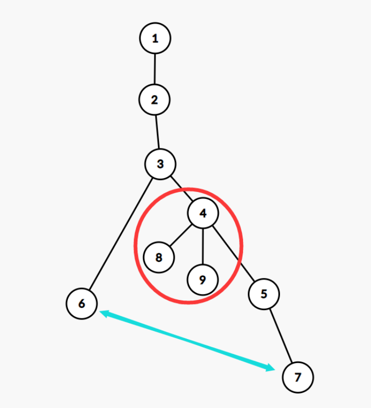
J. 回忆之树

简明题意:

给定一张 n 个点 m 条边的无向连通图 G 以及它的一棵生成树 T。求出所有 u，满足 T 是 G 的一棵以 u 为根的 BFS 树。

考虑在给定的生成树上加边，每加入一条边，只有路径中点的若干子树仍能保持合法（特别的，若中点为 lca，还有 lca 向上的部分），可以用子树加来描述，树上差分即可，时间复杂度 $O(m \log n)$。





K. 骤雨

简明题意:

给定 n 个整型变量的值域 $([l_i, r_i], i \in [1, n])$,
判断是否存在方案能构成 V 图。

前后缀分别贪心维护第 i 个变量的合法最大值 f_i, g_i。

具体地，
$$\begin{cases} f_0 = \infty, & \\ f_i = \begin{cases} 0 & (f_{i-1} - 1 < l_i) \\ \min(f_{i-1} - 1, r_i) & (\text{otherwise}) \end{cases} & (i > 0) \end{cases}$$

判断是否有存在 i 使得 $l_i \le \min(f_i, g_i)$ 即可。

时间复杂度 $O(n)$ 。

L. 交通建造

简明题意:

给 m 棵树，选择添加 $m-1$ 条边得到连通的大树，求大树的直径长度最大值和最小值。

直径 max,min

树的直径

定义: 树上任意两节点之间最长的简单路径。
求法: 从任意点 st 出发找到一个最远点 x, 从 x 出发找到另一个最远点 y。
$dis(x, y)$ 即为直径长度 (直径不唯一, 直径长度唯一)。

max 新图直径: 求出每棵树的直径相加再加 $m-1$
min 新图直径: 求出每棵树的直径的中点，选择其中直径最大的树的直径中点与其它树的直径中点连接形成大树的直径即为最小值。(实际维护最长的 3 个直径长度即可)

M. 计算

简明题意:

表达式求值，运算符为位运算符 ‘|’, ‘&’ 和乘方运算符 ‘^’

维护符号栈和数值栈。根据运算优先级，决定出栈时间。

维护符号栈和数值栈。根据运算优先级，决定出栈时间。
然而，~~Python 一行秒了!~~

```python
print(eval(input().replace("^", "**")))
```

图: Python eval

N. 不为人知的位运算童谣

简明题意:

给定递增序列 $\{a_n\}$, 求最小的 x, 使 $\{a_n \& x\}$ 也是递增序列

考虑**从高位往低位**考虑每一位填 1 还是 0 ，考虑第 i 位是否必须填 1 。
我们先默认第 i 位填 0 ，再**从高位往低位**枚举所有低于 i 的位 j ，此时因为我们如果第 j 位填 1 不会让序列下降就必须填 1 ，这样一定是对的，因为每个能够填上去的 j 都可以去分割序列中的一些段（即固定一些位置的 $a_i \& x < a_{i+1} \& x$ ）
遍历完低于 i 的 j 后，如果填出的数不能让序列递增，则 i 位就必须填 1 了。
时间复杂度： $O(n \log^2 \max\{a\})$

THANKS