2020牛客暑期多校训练营(第九场)

比赛链接

A - Groundhog and 2-Power Representation

Solved by .

题目描述

解题思路

B - Groundhog and Apple Tree

Solved by axforever.

题目描述

给一棵 \$n\$ 个点的树,第一次经过一个点可以回复 \$a_i\$ 点血量,经过一条边消耗 \$w_i\$ 点血量,休息一次恢复 \$1\$ 点血量。初始在 1 号点 \Box hp 为 0。问按某种 dfs 序访问所有点再回到 1 号点最少需要休息几次 \Box \$\sum n \le 10^6\$

解题思路

考虑 dp□设 \$f_i\$ 为在 \$i\$ 号点 hp 为 \$0\$ 时遍历 \$i\$ 的子树需要的休息次数□\$g_i\$ 为 \$i\$ 的子树的 点权 \$-\$ 边权。休息次数可以转化为访问过程中 hp 最小值的相反数。

对于儿子 \$j\$ □将最多花费 \$f_j+w+\max(w-f_j-g_j,0)\$ 点 hp□回复 \$g_j-2w\$ 点 hp □记为 \$(u_j,v_j)\$□问题 在于如何决策访问儿子的顺序。将儿子分为访问过后 hp 增加的和不增加的两组。

首先访问 $v \ge 0$ 的儿子,我们按照 u 递增的顺序访问,维护 hp 的最小值。因为每次访问后 hp 都会增加,因此这样的顺序是最优的。

对于 \$v< 0\$ \square 我们需要按照 \$u+v\$ 递减的顺序访问。简单证明如下:设两点 \$(u_1,v_1)\$ \square \$(u_2,v_2)\$ \square 当前 hp 为 \$c\$ \square 若在不休息的情况下只能先访问 1 后访问 2 ,有 \$u_2\le c+v_1\$ \square \$u_1 > c+v_2\$ \square 即 \$u_2+v_2\le c < u_1+v_1\$ \square —个例子是 hp 为 \$6\$,两对为 \$(5,-3),(4,-1)\$。

时间复杂度 \$O(n\log n)\$

C - Groundhog and Gaming Time

Upsolved by nikkukun.

题目描述

给定 $n \leq 5 \times 10^5$ 个区间 $I_i, r_i \leq 5 \times 10^5$ 个区间 $I_i, r_i \leq 5 \times 10^5$ 有被选中的区间的交集长度的平方和的期望。

注意这里 \$[L, R]\$ 的长度是按 \$R-L+1\$ 算, 出题人不会好好说话, 被坑了。

解题思路

令所有区间端点在数轴的投影将数轴分割成了长度为 s_1 , s_2 , lots, s_m 的 m 个连续的线段,记 $p_i = lots$ $p_i = lots$ 的 $p_i = lots$ 的

 $$$ \left[\left(\sum_{i=1}^m s_i p_i \right)^2 \right] \&= E \left[\sum_{i=1}^m (s_i p_i)^2 + \sum_{i=1}^m (s_i p_i)^2 + \sum_{i=1}^m (s_i p_i)^2 + \sum_{i=1}^m E \left[\sum_{i=1}^m (s_i p_i)^2 + \sum_{i=$

对于前面那部分,如果有 \$k\$ 个原来的线段覆盖了 \$s_i\$□那么在所有 \$2^n\$ 种情况里,只有恰好 \$2^k - 1\$ 种区间选择方法可以让 \$p_i = 1\$ 有贡献 \$\dfrac {2^k - 1} {2^n} \cdot s_i^2\$□否则为 \$0\$。这部分可以直接 \$O(n)\$ 枚举计算。

后面的部分类似,假设有 \$k\$ 个线段恰好能覆盖掉 $$s_i$$ 和 $$s_j$$ 那么只有恰好 $$2^k - 1$$ 种区间选择方法可以让 $$p_i = p_j = 1$$ 有贡献 $$dfrac {2^k - 1}{2^n} \cdot s_j$$ $$g_j$$ $$g_j$$ $$g_j$$

为了不暴力枚举 \$i, j\$ 计算后面的部分,考虑从小到大枚举 \$j\$□一次计算 \$i < j\$ 的所有 \$i\$ 的贡献。假设我们有一个线段树维护了 \$i \in [1, j)\$ 之间,既覆盖了 \$s_j\$ 又覆盖了 \$s_i\$ 的原线段个数 \$k_i\$□那么也可以同时维护 \$\dfrac {2^{k_i} - 1}{2^n} \cdot s_i s_j\$ 之和,单独把那个 \$1\$ 提出来后就能维护。每次从 \$j\$ 移动到 \$j+1\$ 时,更新 \$s_{j+1}\$ 导致的区间 \$k\$ 的增减即可计算答案。

总时间复杂度 \$O(n \log n)\$[]

E - Groundhog Chasing Death

Solved by nikkukun.

题目描述

计算

 $prod {i=a}^b prod {i=c}^d (x^i, y^j) bmod 998244353 $$$

其中 \$0 \leq a, b, c, d \leq 3 \times 10^6\$[]\$1 \leq x, y \leq 10^9\$[]

https://wiki.cvbbacm.com/ Printed on 2025/12/02 17:24

解题思路

为了方便表述,记 \$A = \max\{a, b, c, d\},\ B = \max\{x, y\}\$[]

考虑每个质因子 \$p\$ 对指数的贡献,并令 \$s, t\$ 为使得 \$p^s \mid x\$ 与 \$p^t \mid y\$ 成立的最大整数,则贡献为 \$\$ \sum _{i=a}^b \sum _{i=c}^d \min(si, tj) = \sum _{u=1}^{\infty} \left(\sum _{i=a}^b \underline{u \leq si]\right) \left(\sum _{i=c}^d [u \leq tj]\right) \$\$

这一部分暴力枚举 \$u\$ 可以做到 $O(A \log B)$ 而由于使得 \$s, t > 0\$ 的质数 \$p\$ 只有 $O(\log B)$ 个,故对每个有贡献的质数都计算一次的总时间复杂度为 $O(A \log^2 B)$

另外计算时用容斥把式子拆成四组前缀和的加加减减会好写很多。

F - Groundhog Looking Dowdy

Solved by .

题目描述

解题思路

I - The Crime-solving Plan of Groundhog

Solved by nikkukun.

题目描述

给 n 个数,分成两个部分,每部分没有前导零且是正数,最小化这两个数的乘积并输出[s \sum n \leq n \neq n

解题思路

猜测答案是一位数与剩余能组成的最小的数的乘积,枚举一位数是什么即可。实际上一位数只能是最小的 数,可以不枚举。

J - The Escape Plan of Groundhog

Solved by nikkukun.

题目描述

update: 2020/08/10 2020-2021:teams:i_dont_know_png:multi2020-nowcoder-9 https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:multi2020-nowcoder-9&rev=1597070434

给一个 \$n \times m\$ 的 \$01\$ 矩阵 \$a {n \times m}\$□统计满足条件的子矩阵:

- 1. 子矩阵的四边都是 \$1\$
- 2. 除了四边之外, 子矩阵的 \$0\$ 和 \$1\$ 数量差不超过 \$1\$
- 3. 子矩阵长和宽不小于 \$1\$

\$n, m \leq 500\$∏

解题思路

一般这种题都是枚举一维的上下两个边界,剩下一维做前缀和处理。对于剩下这维,实际是对依次考虑每个满足全\$1\$的左边界,寻找有多少全\$1\$的右边界满足 $$[a_{ij}] = 1] - [a_{ij}] = 0]$的前缀和与它的前缀和差不超过<math>$1$$,这里用类似双指针的方法移动就可以维护相关信息。

总时间复杂度 \$O(n^3)\$□

K - The Flee Plan of Groundhog

Solved by nikkukun.

题目描述

给一个 \$10^5\$ 结点的树□A 在 \$u\$ 且速度是 \$1\$□B 在 \$v = n\$ 且速度是 \$2\$。现在 B 抓 A□A 可以到处逃,问最久可以逃多久。

解颢思路

暴力做,枚举终点 \$x\$□计算 A 和 B 跑过去会不会在中途相遇 (满足 \$2 \cdot \mathrm{dist}(u, x) \leq \mathrm{dist}(v, x)\$ 就不相遇),顺便计算所需时间即可。

赛后总结

nikkukun

- 1. 做题之前一定要先按题意把样例算一下,避免读错题或者是写错算法的情况发生;
- 如果某个算法渐进意义明显超时了,那就不要加优化多次提交了,考虑能不能把复杂度降低一个等级;
- 3. 数量级很大的时候,不要用 map<int, vector<int> >,会卡爆。

qxforever

https://wiki.cvbbacm.com/ Printed on 2025/12/02 17:24

Potassium

From: https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:multi2020-nowcoder-9&rev=1597070434

Last update: 2020/08/10 22:40

