

A GRE

考察对条件语句的运用 (考察小学英语)

其实可以想到不判断所有的字母，只需要判断不相同的字母来减小代码量

"o" "tw" "th" "fo" "fi" "si" "se" "e" "n" "t"

时间复杂度：O(1)

B Words Game

暴力

判断字符串B是不是字符串A的子序列：用pos标记当前字符串B待匹配的字符位置，枚举字符串A，当a[i] == b[pos] 则pos++，如果字符串B的最后一个字符已经被匹配，则字符串B是字符串A的子序列

维护一个ans值，初始为0

对于给出的N个字符串，如果当前ans长度大于等于这个串，Continue（处理下一个串），否则按上面的方法判断，若匹配则更新答案

英文词典长度为m，共有n个待匹配串

时间复杂度：O(mn)

C Arch0n interesting game

规律

将数组a从大到小排序

对于每一个数字a[j] (1<=j<=n)

考虑选取a[i],a[j]，得到的答案是a[j]，这种情况有i-1种

可以得到分段函数

ans = a[i] (1+2+...+i-1<=k && 1+2+...+i>k)

D 另一个简单的游戏

数学

提供者：wy

初始数组为a1,a2...,an

若用枚举的方法做，可以看出 $ans = \sum \lambda_i a_i$ (λ 为加权系数)

由轮换对称性，交换 a_i, a_j ($i \neq j$) 的位置，不影响 ans ，故 $\lambda_i = \lambda$
变形 $ans = \lambda \sum a_i$ (该题 $\lambda = 1/2$)
时间复杂度 ($O(n)$)

E Xiedy And His Password

动态规划

01字符串 s , 长度为 n

$dp[i][j]$: 以 $s[i]$ 结尾，能得到被 3 除余 j 的方案数 ($0 \leq i \leq n-1, 0 \leq j \leq 2$)

从左往右更新 dp 数组

- 若 $s[i]$ 为 '0'，以 $s[i]$ 结尾的数字相当于以 $s[i-1]$ 结尾的数字 * 2

$$3k \cdot 2 = 6 \cdot k \text{ 余 } 0$$

$$(3k+1) \cdot 2 = 6 \cdot k + 2 \text{ 余 } 2$$

$$(3k+2) \cdot 2 = 6 \cdot k + 4 \text{ 余 } 1$$

再考虑 $s[i]$ 独自成数字 0 的情况

故

$$dp[i][0] = dp[i-1][0] + 1$$

$$dp[i][1] = dp[i-1][2]$$

$$dp[i][2] = dp[i-1][1]$$

- 若 $s[i]$ 为 '1'，以 $s[i]$ 结尾的数字相当于以 $s[i-1]$ 结尾的数字 * 2 + 1

$$3k \cdot 2 + 1 = 6 \cdot k + 1 \text{ 余 } 1$$

$$(3k+1) \cdot 2 + 1 = 6 \cdot k + 3 \text{ 余 } 0$$

$$(3k+2) \cdot 2 + 1 = 6 \cdot k + 5 \text{ 余 } 2$$

再考虑 $s[i]$ 独自成数字 1 的情况

故

$$dp[i][0] = dp[i-1][1]$$

$$dp[i][1] = dp[i-1][0] + 1$$

$$dp[i][2] = dp[i-1][2]$$

$$ans = \sum dp[i][0]$$

时间复杂度 $O(n)$

F Operating System

贪心 优先队列

提供者: qkoqhh

其实最主要的问题是在内存满的时候将谁弹出，也就是让留在内存里面的页面尽可能得发挥作用。。

怎么发挥作用？反正碰到相同页面的能发挥作用了嘛。。对内存里面的所有页面找里他下一个相同页面，删除的时候当然是删除下一个页面最远的那个啦。。

然后貌似不太靠谱？可能很多人都考虑到这种情况，例如有2个内存，请求是123122222这样的，把3打入内存时可能觉得保留2也是个不错的方案？我们不妨这么想，第3个以后的2都是上一个2所作出的贡献而不是第1个2，那这样的话内存里面的页面就只能做一次贡献，而做完贡献他们就会被新页面刷新，这样的话显然要优先保留下一个页面离得比较近的了。。。

保证优先级很显然需要用优先队列，按下一页面位置降序排列。。。然后在实现过程中刷新页面这个操作也是直接插入的，旧页面会在队尾不起作用。。。

时间复杂度 $O(n\log n)$

G 小国的复仇

动态规划

提供者：v8

首先，如果我用 x 表示小国， y 表示小杰，则可以直接用二维dp来做 $f(x,y)$ 。复杂度 $O(n^2)$

但是我们可以发现经过 $\times 2$ 的操作后，如果有 x 个小国，那么一定有 $x/2$ 个小杰。

这样我们假设可以计算出一个一维的数组 $g(x)$ 来表示 $f(x,x/2)$ ，然后我们可以发现可以从 $g(x)$ ，通过枚举 x 来计算得到 n 的最小操作数。 $ans = \min(g(x) + (n-x)/(x/2))$;

继续分析我们发现 $g(x)$ 可以由 $g(y)$ 转移而来 $g(x) = g(y) + (x/2 - y)/(y/2) + 1 (y < x)$;

这一步详细解释一下： $f(x,x/2)$ 由 $f(y,k)$ 转移而来（ k 是一个小于 y 的数），但是我们发现有很多状态是重复的，比如 $f(y,k), f(y+k,k)$ 其实对 $f(x,x/2)$ 的更新是一样的(这里先得到 $f(x/2,k)$ 再double一次)。所以我们只需要用所有的 $f(y,y/2)$ 来更新即可。

下面计算一下理论复杂度，因为每一个 $g(x)$ 只会由它的因子转移而来，其它的不能整除 x 的状态是不能更新的。那么我们就用写素数筛的方式写

上限复杂度 $O(n\log(n))$

I Tr0y And His Startup

线段树

提供者：wang9897

这个题主要是理解题意

我们先来考虑一个简单的问题：把攻击的 C 作为每次查询的值给出 那么你需要考虑的是 在查询区间 $[l,r]$ 小于等于 C 的个数是多少

这个问题可以通过线段树维护最小值什么的解决(这并不是这个题的重点)

然后考虑到这题的C是作为[1,C]的离散随机变量给出 也就是对于每个攻击c在每次出现的概率都是1/C 即在每次ci的攻击的时候 小于等于ci的值都会对答案产生贡献 所以对于期望而言(期望可百度概率论什么的) 就转化为

$$\sum (\text{num}(\text{小于等于}ci\text{的个数}) * ci) / C \quad (l \leq i \leq r)$$

即为区间查询所求

然而这个式子是没办法解决的 因为题目说在每次攻击后 是区间修改的 也就是我们这个式子需要再变形 转化成

$$\sum ((xi+C) * (C-xi+1)) / 2 * C \quad (xi\text{是该位置的权值,且}l \leq i \leq r)$$

显然我们需要将求和里面的式子展开

$$\sum (-xi^2 + xi + C^2 + C)$$

做到这里 就是很基础的线段树操作了 我们只需要维护区间xi的和 以及 区间长度 (打lazy标记就能处理) 就可以进行更新操作 需要注意的是 操作都在mod下进行的 所以我们要求(2*C)的逆元 然后这个问题就圆满解决了

时间复杂度(nlogn)

J Database

模拟

这题就是大模拟，考察对字符串的处理和一些容器（数据结构）的应用，细节很多