

Universidade de São Paulo

Instituto de Matemática e Estatística

IX Simulado dos Bixos

Editorial

Problema	Autor	Preparado por
A - Acerto de contas	Thiago Oliveira	Otávio de Oliveira
B - Batalha Espacial	Rafael Sesoko	Paulo Dias
C - Candidatos cósmicos	Rafael Sesoko	Rafael Sesoko
D - Dividindo a pizza solar	Enrique Junchaya	Enrique Junchaya
E - Escavando Mercúrio	Enrique Junchaya	Gabriel Morete
F - Festa Lunina	Marcelo Lage	Cauê da Costa
G - Genealogia alienígena	Marcelo Lage	Marcelo Lage
H - Homenagem a Saturno	Pedro Simões	Pedro Simões
I - Investigando Marte	Marcelo Lage	Nathan Martins
J - Jantar em Jupiter	Thiago Oliveira	Thiago Oliveira

Table 1: Problemas da prova.

A. Acerto de contas.

Tags : implementação.

Seja D a maior diferença entre dois números de celas adjacentes na ordem circular (se os c_i estão em ordem crescente, são os valores $c_{i+1} - c_i$ para $1 \leq i \leq n - 1$ e $c_1 - c_n + 360$). Se o intervalo de celas vigiadas é estritamente menor que $360 - (D - 1)$, então existe um intervalo de pelo menos D celas não vigiadas, e entre essas D celas existe alguma ocupada. Logo, é necessário que pelo menos $361 - D$ celas sejam vigiadas pelas câmeras.

B. Batalha espacial.

Tags : ad hoc.

Dado um inteiro n , o problema pede apenas para imprimir um inteiro estritamente maior que n . Basta imprimir $n + 1$.

C. Candidatos cósmicos.

Tags : matemática.

Como o Coach deve convencer a maioria dos membros de um time, ele deve convencer exatos dois membros por time. Note que a quantidade de times é $\frac{n}{3}$. Logo a quantidade de membros que o Coach deve convencer deve ser $\frac{2n}{3}$.

D. Dividindo a pizza solar.

Tags : guloso, ordenação.

Ordene os preços dos sabores para supor que $p_1 \geq p_2 \geq \dots \geq p_n$. É fácil ver que a i -ésima pizza mais cara terá preço pelo menos p_{2i+1} , pois não é possível usar todos os $2i + 1$ sabores mais caros nas i primeiras pizzas, que só tem $2i$ metades ao todo. Logo, a resposta é pelo menos $p_1 + p_3 + \dots + p_{2n-1}$, que pode ser atingida colocando os sabores p_{2i-1} e p_{2i} na i -ésima pizza.

E. Escavando Mercúrio.

Tags : guloso, implementação.

Como a altura de um monte não pode aumentar, a altura de todos os montes ao final não pode ser maior que $m := \min_{1 \leq i \leq n} a_i$. Assim, devemos tirar pelo menos $a_i - m$ metros do monte i , e a resposta final é $\sum_{i=1}^n (a_i - m)$, que pode ser calculado em $O(n)$.

F. Festa lunina.

Tags : equações lineares, matemática.

Seja x a quantidade de pessoas que pagaram inteira, e y a quantidade de pessoas que pagaram meia. Temos então o seguinte sistema de equações:

- $x + y = p$
- $vx + \frac{v}{2}y = a$

Substituindo o valor de x por $p - y$, e resolvendo para o valor de y , temos que $y = 2p - 2a/v$.

G. Genealogia alienígena.

Tags: matemática.

Para n dado, queremos encontrar o número de soluções para $a + ar + \dots + ar^m = n$, sendo a, r, m inteiros com $a \geq 1, r > 1, m \geq 0$. Reescrevendo como $a = \frac{n}{1+r+\dots+r^m}$, vemos que o valor que queremos encontrar é o número de pares $(r > 1, m \geq 0)$ de inteiros com $1 + r + \dots + r^m$ divisor de n . $m = 0$ corresponde a uma única solução.

A primeira ideia é iterar por r de 1 a n e por m de 1 até algum valor que faça a soma ser maior que n e verificar, mas essa solução fica $O(n \log n)$, que não vai passar.

O refinamento da ideia é fazer isso para $m \geq 2$, pois $m = O(\log n)$ e $1 + r + r^2 > n$ para $r > \sqrt{n}$, e essa parte fica $O(\sqrt{n} \log n)$.

Para $m = 1$, queremos simplesmente encontrar a quantidade de divisores positivos de n que são pelo menos 3, e isso pode ser feito em $O(\sqrt{n})$ percorrendo todos os divisores.

A solução final fica $O(\sqrt{n} \log n)$.

H. Homenagem a Saturno.

Tags: strings.

Podemos representar o cordão dado como uma sequência $[c_1, c_2, \dots, c_m]$ de inteiros correspondendo ao tamanho de sequências monocromáticas contíguas maximais no cordão. Por exemplo, o cordão “abbbbaa” corresponde à sequência $[1, 3, 2]$. Para calcular a resposta final, temos duas opções: a sequência de asteroides pode corresponder a apenas dois elementos consecutivos da sequência c_i (ou seja, $\max_{1 \leq i \leq m-1} \min(c_i, c_{i+1})$) ou pode ser formada por três elementos consecutivos da sequência

c_i (ou seja, $\max_{c_i \leq c_{i-1} + c_{i+1}} c_i$), como no caso do cordão “abbbbaa” descrito no enunciado. A solução fica em $O(n)$.

I. Investigando Marte.

Tags: ad-hoc, implementação.

Para detectar quando o robô para sua viagem, não é suficiente detectar quando ele visita novamente uma casa pela qual ele já passou: ele precisa passar por uma casa que já passou orientado da mesma forma. Assim, vamos simular o caminho do robô mantendo sua posição e sua direção atuais e a cada passo atualizamos um vetor booleano tridimensional $vis[n][m][4]$ de acordo com a posição e direção atuais do robô. Em seguida, tentamos andar na mesma direção que a atual. Se isso não é possível (ou seja, a próxima casa é ‘#’ ou está fora do tabuleiro), trocamos de posição e continuamos o laço. Cada iteração demora $O(1)$ e fazemos no máximo $4nm$ iterações, logo a complexidade final é $O(nm)$.

J. Jantar em Jupiter.

Tags: sliding window, two pointers.

Iremos mostrar duas soluções.

A primeira é por busca binária na resposta usando sliding window. Queremos encontrar o maior m tal que algum intervalo de tamanho m da sequência tem no máximo k elementos distintos (se m funciona, qualquer $m' < m$ também funciona). Vamos começar fixando a janela de tamanho m nos primeiros elementos da sequência e a cada iteração iremos “mover” essa janela para a direita, mantendo o número de elementos distintos nessa janela (isso pode ser feito usando um vetor de frequência: quando uma frequência cai para zero, o número de elementos distintos diminui em um; quando uma frequência aumenta para um, o número de elementos distintos aumenta em um). Assim, em tempo $O(n)$ podemos dizer se m funciona. A resposta fica em $O(n \log n)$ usando busca binária.

A segunda solução também usa vetor de frequência com *two pointers* (um inteiro para indicar o início do intervalo atual e outro inteiro para o fim desse intervalo): começamos com um intervalo de tamanho 1 contendo apenas o primeiro elemento da sequência mantendo a quantidade de elementos distintos nele e vamos mover a extremidade *direita* do intervalo mantendo seu tamanho até que haja $k + 1$ elementos distintos. Então vamos mover a extremidade *esquerda* do intervalo até que haja k elementos distintos no intervalo novamente. Repetimos o processo com o intervalo atual até que a extremidade direita seja a última posição da sequência. Sempre que tivermos em mãos uma sequência com no máximo k elementos distintos atualizamos o tamanho do nosso intervalo maximal. Essa solução fica em $O(n)$.