

Universidade de São Paulo

Instituto de Matemática e Estatística

VIII Simulado dos Bixos

Editorial

Problema	Autor	Preparado por
A - Ágoranegócio	Marcelo Lage	Marcelo Lage
B - Balões Quânticos	Matheus Paulo	Willian Mori
C - Convite dos Deuses	Gabriel Morete	Gabriel Morete
D - Dupla Sertanejo	Enrique Junchaya	Enrique Junchaya
E - Enigma da Esfinge	Pedro Sousa	Pedro Sousa
F - Fraude na votação	Marcelo Lage	Nathan Luiz
G - Gusteseu e a Maynotauro	Matheus Paulo	Matheus Paulo
H - Herói Morethor	Thiago Oliveira	Otávio de Oliveira
I - Ideias Primordiais	Thiago Oliveira	Thiago Oliveira
J - Jantar dos Deuses	Enrique Junchaya	Pedro Simões

Table 1: Problemas da prova.

A. Ágoranegócio.

Tags : Matemática.

Vamos resolver o problema para um único mercado. Note que o objetivo de Renang e Wan pode ser reescrito como comprar uma quantidade total par de frutas nesse mercado.

Se a mesma venda v vende tanto pacotes com quantidade par como pacotes com quantidade ímpar de frutas, a resposta será “SIM” (basta eles comprarem quaisquer pacotes nas outras vendas e, na venda v , escolhem um pacote que deixa a soma total dos tamanhos dos pacotes par).

Caso contrário, todas as vendas vendem pacotes ou todos de tamanho par ou todos de tamanho ímpar. Daí, basta verificar se o número de vendas que vendem apenas pacotes de tamanho ímpar é par (resposta “SIM”) ou é ímpar (resposta “NAO”).

A complexidade dessa solução é $O(p)$ se p é o total de pacotes em todos os mercados.

B. Balões Quânticos.

Tags : Guloso, estruturas de dados

Observamos que para atingirmos o tempo mínimo necessário para que a quantidade de balões inflados seja menor que S , uma estratégia intuitiva que garante atingir o tempo mínimo é sempre olhar para a cor de balão com maior quantidade. Esse tipo de estratégia é chamada de estratégia gulosa (pois olhamos apenas para o estado atual — no caso, o número de balões de cada cor — para

decidir o que faremos). Para isso, existem as estruturas de dados *priority_queue* e *multiset*, que são capazes de dado uma coleção de números (que pode ser modificada), extraímos o maior número.

C. Convite dos Deuses.

Tags : Implementação, força bruta.

Força bruta: podemos testar cada possível subseção retangular da mesa diretamente. São $O((nm)^2)$ tais subseções, e a verificação se cada uma delas é composta apenas por bolinhas de queijo pode ser feita em $O(nm)$, verificando uma a uma (é possível fazer isso de forma mais eficiente em geral, mas nesse problema isso não é necessário).

A complexidade total é $O((nm)^3)$, que é suficiente para passar pois $nm \leq 400$.

D. Dupla Sertanejo.

Tags : Implementação, força bruta.

Note que se fixamos os primeiros dois números da sequência, o resto da sequência já fica determinado. Temos apenas 9 possibilidades para os dois primeiros números e de fato duas delas não são válidas (é impossível começar com “0 0” ou com “2 2”). Portanto, basta provar todas as possibilidades e verificar se podem casar com a sequência dada.

A forma mais simples de fazer isso é usando dois laços **for**: o primeiro para fixar o primeiro número e o segundo para fixar o segundo número. Dentro dos laços, podemos construir a sequência única determinada pelos dois números fixados e comparar ela com a sequência dada. Se for possível obter a sequência da entrada apagando alguns dígitos da sequência construída, então devemos adicionar +1 à nossa resposta.

E. Enigma da Esfinge.

Tags : Matemática, ad-hoc.

Podemos reescrever a situação final que queremos obter como $\frac{f}{v+f} = \frac{3}{4}$ ou, ainda, $3v = f$.

Se inicialmente $3v - f < 0$, a frase será verdadeira e a frase será proclamada verdadeira até que $3v - f \geq 0$, o que acontecerá $\left\lceil \frac{f-3v}{3} \right\rceil$ vezes, pois a cada vez ela aumenta o valor de $3v - f$ em 3.

Logo, basta resolver para quando os valores iniciais de v e f são tais que $3v - f \geq 0$. Então a resposta será exatamente $3v - f$, pois a frase “mais que 75% das estatísticas são falsas”, ao ser proclamada, será falsa, e $3v - f$ decrescerá em 1, até chegar em 0.

F. Fraude na votação.

Tags : Busca binária.

A ideia é fazer busca binária na resposta. Em outras palavras, considere a propriedade

$P(N)$: Kobus consegue ganhar comprando N votos

É claro que se vale $P(N)$ para um certo N , $P(M)$ vale para todo $M > N$. Logo, se queremos descobrir o menor N tal que $P(N)$ vale, podemos fazer busca binária em N , em até $O(\log(10^9 m))$ passos.

Assim, podemos nos perguntar: “dada uma quantidade x de votos, Kobus consegue ganhar comprando no máximo x votos?”

E isso é fácil de verificar em $O(m)$, pois basta

$$\sum_{i=1}^n \max(0, a_i - (a_0 + x - 1)) \leq x$$

A solução completa roda em $O(m \log(10^9 m))$.

G. Gusteseu e a Maynotauro.

Tags: Matemática.

Um fato importante é que o número de maneiras de percorrer um tabuleiro $m \times n$ a partir do seu canto superior esquerdo, até o canto inferior direito, utilizando a menor quantidade de passos possível (ou seja, apenas passos do tipo $\downarrow$ e $\rightarrow$), é igual a $\binom{n+m-2}{n-1}$.

Isso é verdade pois cada um desses caminhos pode ser visto como um anagrama de $\underbrace{\downarrow \cdots \downarrow}_{m-1} \underbrace{\rightarrow \cdots \rightarrow}_{n-1}$.

Daí, a quantidade de caminhos em um tabuleiro saindo de $(1, 1)$ até (m, n) sem passar por (x, y) é igual a $\binom{n+m-2}{n-1}$ menos o número de caminhos que passam por (x, y) . Mas, para passar por (x, y) , o conjunto dos primeiros $x + y - 2$ movimentos está determinado (assim como os últimos $n + m - x - y$, e portanto esse número será $\binom{x+y-2}{x-1} \cdot \binom{n+m-x-y}{n-x}$).

Para os binomiais, é possível precalculá-los em uma matriz usando a relação de Stifel e consultar depois em $O(1)$.

H. Herói Morethor.

Tags: Implementação.

É importante observar que Morethor enfrenta os monstros (possivelmente nenhum) na ordem em que são encontrados, ou seja, na ordem dada pela entrada, logo ordenar os valores de F_i é uma abordagem errada. Assim, podemos manter uma variável X que armazena a força atual de nosso herói, inicialmente com a habilidade H informada. Podemos então iterar por cada um dos valores das forças dos monstros na ordem em que foram dados, verificando na i -ésima iteração se o valor atual da força de Morethor é maior ou igual que F_i , ou seja, $X \geq F_i$. Caso seja, atualizamos X realizando $X = X + F_i$, ou seja, acumulamos o valor da força do monstro à força do herói, e prosseguimos. Caso contrário, a resposta é “NAO”. Se chegamos ao final da sequência de números e a condição foi respeitada, a resposta é “SIM”.

I. Ideias Primordiais.

Tags: Ad-hoc, implementação.

Note que, dada uma palavra da tábula, existe *no máximo uma* palavra de U que possa corresponder a ela. Por exemplo, PWP no caso 1 só poderia corresponder a OVO (não a VOO nem a TLE).

Dessa forma, a ideia é manter a permutação final considerando que cada nova palavra lida “força” alguns valores nessa permutação. No caso 2 da folha, por exemplo, inicializamos com:

attempt permutation = “\$”

onde \$ representa uma posição ainda não forçada.

BD só pode corresponder a AC, logo atualizamos *attempt permutation*:

attempt permutation = “B\$D\$”

CMOR só pode corresponde a AMOR, mas a posição já está ocupada com uma letra *diferente* da que queremos colocar lá agora. Concluimos que não existe a permutação.

No caso 4 da folha, teríamos após ler todas as strings:

attempt permutation = “A\$\$\$\$\$\$\$\$\$\$\$\$OP\$\$ST\$\$\$\$\$”

Agora basta apenas verificar se é possível preencher as posições \$ não reveladas de forma que a string total seja uma permutação.

Existem vários jeitos de fazer isso, por exemplo com *set*: mantendo um *set* com todas as letras do alfabeto, inicialmente iteramos para tirar dele todas as letras que já tem posição forçada na permutação (e nessa iteração também verificamos se alguma letra aparece duas vezes, o que significa que não há a permutação procurada) e logo depois iteramos para preencher as posições marcadas com \$ com os elementos restantes do set.

J. Jantar dos Deuses.

Tags: Matemática.

De acordo com o enunciado, cada refeição no bandeirão custa duas dracmas, então, se houverem sido gastas d dracmas, é possível inferir que foram realizadas $\frac{d}{2}$ refeições. Observa-se, ainda, que, como x e y têm sempre a mesma paridade, $d = x - y$ é sempre divisível por 2. A partir disso, sendo x o valor da recarga e y o saldo atual, a quantidade de refeições realizadas é dada por $\frac{d}{2} = \frac{x-y}{2}$.