Algoritmos

Yvens Ian

29ª Semana Olímpica - 30 de janeiro de 2026

---

1 O que é um Algoritmo?

Definição 1. Um problema computacional P é definido como um par ordenado de conjuntos (I, O), no qual I é o conjunto de entrada, e para toda instância do problema $e \in I$, o conjunto $O(e)$ são as possíveis saídas (respostas) dessa instância.

Exemplo 2. Se o problema P é "Calcule o máximo divisor comum de a e b para $a, b \in \mathbb{Z}_{>0}$", então, $I = \mathbb{Z}_{>0} \times \mathbb{Z}_{>0}$ e $O(a, b) = \{\text{mdc}\,(a, b)\}$.

Exemplo 3. Se o problema P é "Encontre um número inteiro menor que $a \in \mathbb{Z}_{>0}$", então, $I = \mathbb{Z}_{>0}$ e $O(a) = \{1, 2, 3, \ldots, a-1\}$.

Definição 4. Um algoritmo A é uma sequência de instruções bem definida e finita que recebe uma entrada e e responde uma saída $A(e)$.

Uma forma clássica de pensar em um algoritmo é como uma receita de bolo. O algoritmo recebe ingredientes (farinha, açúcar, ovos, etc), a entrada, e retorna um bolo, a saída.

Definição 5. Dizemos que o algoritmo A resolve o problema $P = (I, O)$ se para toda instância da entrada $e \in I$, temos que $A(e) \in O(e)$, ou seja, para toda entrada possível, A retorna uma saída correta.

2 Meu Algoritmo é ótimo?

Na computação, geralmente nos preocupamos com a complexidade de um algoritmo. Ou seja, a quantidade de passos que temos que dar até chegar na resposta.

Na maioria dos problemas de matemática, não olhamos tanto para isso, tendo em vista que na maioria dos casos queremos apenas apresentar um algoritmo que resolve o problema, ou usar um algoritmo pra provar algo, e não otimizar esse algoritmo para fazer poucos passos. Porém, também existem problemas que limitam a quantidade de passos que podemos dar.

Na computação usamos uma notação especial para calcular a complexidade de um algoritmo, chamada de Grande-O, ou O-Grande, ou Ozão.

Definição 6. Sejam f e g duas funções definidas no mesmo subconjunto dos números reais. Dizemos que

$$f = O(g)$$

se, e somente se, existe um real positivo M e um número real x_0 tal que

$$|f(x)| \leq M|g(x)| \text{ para todo } x \geq x_0.$$

Exemplo 7. Mesmo que no geral $2x^2 > x^2+x$, veja que $2x^2 = O(x^2+x)$ e que $x^2+x = O(2x^2)$.

Exercício 1. *Coloque as seguintes funções em ordem de O, ou seja, da esquerda para direita de modo que se $f_1 = O(f_2)$, então, f_1 vem antes de f_2:*

$$2^x, x^{1.3}, x^x, x^2, x!, e^{x^2}, x, \ln x, \ln x^2, \ln^2 x.$$

Usamos isso, pois, na computação, o que mais importa é a velocidade de crescimento do número de passos, não sendo sempre necessário calcular com exatidão essa quantidade. Mas existem exemplos nos quais precisamos calcular esse valor não só assintoticamente.

3 Algoritmos Gulosos

Definição 8. Um algoritmo guloso é qualquer algoritmo que segue a escolha localmente ótima do problema a cada passo.

Por exemplo, para achar n objetos satisfazendo condições dadas, você pode pegar um de cada vez sem causar problemas. Depois olhar a situação na qual não é possível adicionar mais.

Exemplo 9 (CSES Permutation Prime Sums)**.** Dado um inteiro positivo n, mostre que é possível criar uma permutação A de $\{1, 2, \ldots, n\}$ de modo que $A_i + i$ seja sempre um número primo.

Exemplo 10 (IMO 2014)**.** Um conjunto de retas no plano está em posição geral se não há duas retas paralelas, nem três concorrentes. Um conjunto de retas em posição geral corta o plano em regiões, das quais algumas têm área finita; chamamos essas de regiões finitas. Prove que para todo n suficientemente grande, em qualquer conjunto de n retas em posição geral é possível colorir pelo menos $\sqrt{n}$ retas de azul de modo que nenhuma das regiões finitas possua uma borda completamente azul.

Em vários problemas a estratégia gulosa não produz a solução ótima. Daí precisamos ajustar o guloso, ou seguir para estratégias mais globais.

Exemplo 11 (OBM 2016)**.** Qual a maior quantidade de inteiros positivos menores ou iguais a 2016 que podemos escolher de modo que não existem dois que diferem por 1,2, ou 6?

4 Algoritmos Clássicos

Exemplo 12 (Busca Binária)**.** Escreva um algoritmo em $O(\log(n))$ que recebe uma lista estritamente crescente de inteiros $A = \{a_1, a_2, \ldots, a_n\}$ e um inteiro x e retorne o primeiro índice i tal que $a_i \geq x$.

Exemplo 13 (Merge Sort)**.** Escreva um algoritmo em $O(n \log(n))$ que recebe uma lista de números inteiros e retorne essa lista ordenada da esquerda para direita.

Exemplo 14 (Exponenciação Rápida)**.** Escreva um algoritmo em $O(\log(n))$ que recebe dois números inteiros a, n e calcula a^n.

Exemplo 15 (Crivo de Eratóstenes)**.** Escreva um algoritmo em $O(n \log(\log(n))$ que recebe um números inteiros n e retorna a lista dos primos de 1 até n.

Exemplo 16 (Algoritmo de Euclides)**.** Escreva um algoritmo em $O(\log \min a, b)$ que recebe dois números inteiros a, b e retorna a $\operatorname{mdc} a, b$.

5 Problemas!!!

Problema 1. *Você possui n moedas idênticas das quais $n-1$ possuem o mesmo peso e 1 é mais pesada que as demais. Você também tem uma balança de comparação, ou seja, consegue colocar dois conjuntos de objetos, um em cada prato, e checar se um é mais pesado que o outro, ou se possuem o mesmo peso. Qual a mínima quantidade de pesagens necessárias para encontrar qual a moeda mais pesada?*

Problema 2 (Torneio das Cidades)**.** *a) Existem $2n+1$ baterias ($n > 2$). Não sabe-se quais funcionam, mas sabe-se que o número de baterias boas é maior que o número de baterias ruins. Uma lâmpada utiliza duas baterias e só funciona se ambas forem boas. Qual é o menor número de tentativas suficientes para fazer a lâmpada funcionar?*

b) Mesmo problema, mas com $2n$ baterias ($n > 2$) das quais n são boas e n são ruins.

Problema 3 (Rússia 2011)**.** *Em uma matriz $2 \times n$ temos números reais positivos tais que a soma dos números em cada uma das n colunas é igual a 1. Mostre que é possível escolher um número em cada coluna de modo que a soma dos números escolhidos em cada linha seja, no máximo, $\dfrac{n+1}{4}$.*

Problema 4 (Rússia 2014)**.** *Existem n células com índices de 1 a n. Inicialmente, em cada célula há um cartão com o índice correspondente. Vasya rearranja os cartões de modo que, na i-ésima célula, passe a haver um cartão com o número a_i.*

Petya pode trocar quaisquer dois cartões com números x e y, mas para isso ele precisa pagar $2|x-y|$ moedas. Mostre que Petya consegue devolver todos os cartões às suas posições originais pagando, no máximo,

$$|a_1 - 1| + |a_2 - 2| + \cdots + |a_n - n|$$

moedas.

Problema 5 (OMpD 2024)**.** *Lavidópolis é uma cidade com 2024 bairros. Lavi Dopes foi eleito prefeito e, como ele viu que não havia nenhuma estrada na cidade, pediu para Gil Bento, o engenheiro monstrão, projetar as estradas da cidade de acordo com as seguintes regras:*

1. *quaisquer dois bairros estão ligados por no máximo uma estrada de mão dupla;*
2. *para quaisquer dois bairros, existe exatamente uma rota de um bairro a outro, rota essa que pode passar por alguns bairros intermediários, mas nunca passa por um mesmo bairro mais de uma vez.*

O prefeito Lavi Dopes quer tentar a reeleição, mas como ele não sabe nada sobre a cidade e só aparece em tempos de campanha (ele passou esse tempo todo roub... quer dizer, pensando em problemas de matemática), ele deseja descobrir algum par de bairros tal que a quantidade de estradas que faz parte da rota que os conecta é máxima, dentre todos os pares de bairros. Para isso, ele começa a fazer diversas perguntas a Gil Bento, todas elas da seguinte maneira: ele escolhe dois dos 2024 bairros, digamos A e B, e pergunta:

> *"Dados os bairros A e B, quantas estradas fazem parte da rota conectando A até B?"*

Sabendo que Gil Bento sempre responde corretamente a cada pergunta, determine o número mínimo de perguntas que Lavi Dopes precisa fazer para alcançar seu objetivo, não importando como Gil Bento tenha projetado as estradas de Lavidópolis.

Problema 6 (Vingança Olímpica 2024)**.** *Davi e George estão fazendo um city tour por Fortaleza, e Davi começa liderando. Fortaleza é uma cidade que se organiza como um tabuleiro $n \times n$. Eles escolhem uma casa do tabuleiro $n \times n$ para começar, e podem se mover de uma casa para outra desde que ambas compartilhem uma aresta. Essa aresta é chamada de rua (para cada duas casas vizinhas no tabuleiro, há uma rua entre elas).*

Algumas ruas são perigosas. Se Davi e George passam por uma rua perigosa, eles ficam com medo e trocam quem lidera o city tour. O objetivo de ambos é passar por todas as casas de Fortaleza exatamente uma vez. Mas, se o city tour acaba com George liderando, o mundo inteiro fica desempregado e todos morrem de fome.

Dado que existe uma rua que não é perigosa, prove que Davi e George conseguem cumprir seu objetivo sem que todos morram de fome.

Problema 7 (USAMO 2011)**.** *Na nação de Onewaynia, certos pares de cidades estão conectados por estradas. Cada estrada conecta exatamente duas cidades (as estradas podem se cruzar entre si, por exemplo por meio de pontes). Algumas estradas possuem capacidade de tráfego igual a 1 unidade e outras possuem capacidade de tráfego igual a 2 unidades. Além disso, em cada estrada o tráfego só pode ocorrer em um único sentido.*

Sabe-se que, para toda cidade, a soma das capacidades das estradas incidentes a ela é sempre ímpar. O ministro dos transportes precisa atribuir uma direção a cada estrada. Prove que ele pode fazê-lo de tal maneira que, para toda cidade, a diferença entre a soma das capacidades das estradas que entram na cidade e a soma das capacidades das estradas que saem da cidade seja sempre exatamente igual a 1.

Problema 8 (OMpD 2021)**.** *Branca de Neve possui, em seu enorme cesto, 2021 maçãs, e ela sabe que exatamente uma delas possui um veneno mortal, capaz de matar um ser humano horas após ingerir apenas um mísero pedaço. Ao contrário do que dizem os contos de fadas, Branca de Neve é mais malévola do que a Rainha Má, e não se importa com a vida dos sete anões. Por isso, ela decidiu usá-los para descobrir qual maçã é envenenada.*

Para tanto, no começo de cada dia, Branca de Neve prepara algumas saladas de maçã (cada salada é a mistura de pedaços de algumas maçãs escolhidas por ela) e obriga alguns dos anões (possivelmente todos) a comer uma salada cada um. No fim do dia, ela observa quem morreu e quem sobreviveu, e no dia seguinte ela novamente prepara algumas saladas de maçã, obrigando alguns dos anões sobreviventes (possivelmente todos) a comer uma salada cada um. Ela continua a fazer isso, dia após dia, até descobrir a maçã envenenada ou até que todos os anões morram.

(a) Prove que existe uma estratégia na qual Branca de Neve consegue descobrir a maçã envenenada depois de alguns dias.

(b) Qual é o número mínimo de dias que Branca de Neve necessita para descobrir a maçã envenenada, independentemente da sorte que ela tenha?

Problema 9 (OBM 2018)**.** *Considere $4n$ pontos no plano, com a propriedade de que nenhum três deles são colineares. Usando esses pontos como vértices, formam-se $\binom{4n}{3}$ triângulos. Mostre que existe um ponto X do plano que pertence ao interior de pelo menos $2n^3$ desses triângulos.*

Problema 10 (IMO 2010)**.** *Cada uma das seis caixas $B_1, B_2, B_3, B_4, B_5, B_6$ contém inicialmente uma moeda. As seguintes operações são permitidas:*

1. *Escolher uma caixa não vazia B_j, $1 \leq j \leq 5$, remover uma moeda de B_j e adicionar duas moedas a B_{j+1};*

2. *Escolher uma caixa não vazia B_k, $1 \leq k \leq 4$, remover uma moeda de B_k e trocar o conteúdo (possivelmente vazio) das caixas B_{k+1} e B_{k+2}.*

Determine se existe uma sequência finita de operações dos tipos permitidos tal que as cinco caixas B_1, B_2, B_3, B_4, B_5 fiquem vazias, enquanto a caixa B_6 contenha exatamente $2010^{2010^{2010}}$ moedas.

Problema 11 (IMO 2017)**.** *Um inteiro $N \geq 2$ é dado. Um total de $N(N+1)$ jogadores de futebol, dos quais não há dois de mesma altura, ficam em fila. Sir Alex quer remover $N(N-1)$ jogadores dessa fila deixando uma nova fila de $2N$ jogadores satisfazendo as N condições:*

1) Ninguém fica entre os dois maiores jogadores;

2) Ninguém fica entre o terceiro e o quarto maiores jogadores;

$\vdots$

3) Ninguém fica entre os dois menores jogadores.

Problema 12 (IMOSL 2019)**.** *Alice possui um mapa do País das Maravilhas, um país que consiste de $n \geq 2$ cidades. Para cada par de cidades, existe uma estrada estreita ligando uma cidade à outra. Certo dia, todas as estradas passam a ser declaradas como sendo de "mão única". Alice não tem nenhuma informação sobre a direção das estradas, mas o Rei de Copas se oferece para ajudá-la. Ela pode fazer a ele um certo número de perguntas. Em cada pergunta, Alice escolhe um par de cidades e o Rei de Copas informa a direção da estrada que conecta essas duas cidades.*

Alice quer saber se existe ao menos uma cidade em País das Maravilhas com, no máximo, uma estrada saindo dela. Prove que ela sempre consegue descobrir isso fazendo no máximo $4n$ perguntas.

6 Problemas IOI

By Caique Paiva e Maria Clara Fontes Silva

O tempo limite de cada problema é de alguns segundos, então evite fazer um número grande demais de operações (algo maior que 10^8), por exemplo, se n vai até 10^5, não faça um algoritmo com $O(n^2)$ operações básicas.

Como são problemas de programação, seus estilos são um pouco diferente, mas todos se resumem a resolver uma tarefa criando um algoritmo eficiente e correto. Aos que quiserem se aventurar, todos os problemas possuem um link para o CodeForces, onde um código pode ser enviado, para verificar a corretude do algoritmo.

Problema 13 (IOI 2002 - Utopia Divided)**.** *A bela terra de Utopia foi uma vez devastada pela guerra. Quando as hostilidades diminuíram, o país foi dividido em quatro regiões por uma longitude (linha norte-sul) e uma latitude (linha leste-oeste). A interseção dessas linhas ficou conhecida como o ponto $(0, 0)$. Todas as quatro partes reivindicaram o nome Utopia, mas com o passar do tempo elas se tornaram geralmente conhecidas como Utopia 1 (nordeste), 2 (noroeste), 3 (sudoeste) e 4 (sudeste). Um ponto em qualquer uma das regiões era identificado pela sua distância a leste e sua distância ao norte de $(0, 0)$. Essas distâncias podiam ser negativas; portanto, um ponto em Utopia 2 era designado por um par (negativo, positivo), em Utopia 3 por um par (negativo, negativo), em Utopia 4 por (positivo, negativo) e em Utopia 1 por um par de números positivos.*

Utopia 2	*Utopia 1*
$(-,+)$	$(+,+)$
Utopia 3	*Utopia 4*
$(-,-)$	$(+,-)$

Um grande problema era que os cidadãos não tinham permissão para cruzar fronteiras. Felizmente, alguns engenhosos competidores da IOI de Utopia desenvolveram um meio seguro de teletransporte. A máquina requer números de código, cada um dos quais só pode ser usado uma vez. Agora, o desafio enfrentado pela equipe, e por você, é guiar o teletransportador de sua posição inicial de $(0,0)$ *para as regiões de Utopia na ordem solicitada. Você não se importa onde em uma região você aterrissa, mas você terá uma sequência de* N *números de região que especificam as regiões nas quais o teletransportador deve aterrissar. Pode ser solicitado que você pouse na mesma região em duas ou mais paradas consecutivas. Depois de deixar o ponto inicial* $(0,0)$*, você nunca deve pousar em uma fronteira.*

Você receberá como entrada uma sequência de $2 \cdot N$ *números de código e deve escrevê-los como uma sequência de* N *pares de código, colocando um sinal de mais ou um sinal de menos antes de cada número. Se você está atualmente no ponto* (x, y) *e usa o par de código* $(+u, -v)$*, você será teletransportado para o ponto* $(x + u, y - v)$*. Você tem os* $2 \cdot N$ *números, e você pode usá-los em qualquer ordem que desejar, cada um com um sinal de mais ou um sinal de menos.*

Suponha que você tenha números de código 7, 5, 6, 1, 3, 2, 4, 8 e deve guiar o teletransportador de acordo com a sequência de números de região 4, 1, 2, 1. A sequência de pares de código $(+7,-1),(-5,+2),(-4,+3),(+8,+6)$ *consegue isso, pois teletransporta você de* $(0,0)$ *para os locais* $(7,-1),(2,1),(-2,4)$ *e* $(6,10)$ *nessa ordem. Esses pontos estão localizados em Utopia 4, Utopia 1, Utopia 2 e Utopia 1, respectivamente.*

Você recebe $2 \cdot N$ *números de código distintos e uma sequência de* N *números de região indicando onde o teletransportador deve pousar. Construa uma sequência de pares de código a partir dos números dados que guiem o teletransportador para passar pela sequência de regiões dada.*

Problema 14 (IOI 2018 - Mechanical Doll)**.** *Uma boneca mecânica é uma boneca que repete automaticamente uma sequência específica de movimentos.*

No Japão, muitas bonecas mecânicas foram criadas desde tempos antigos. Os movimentos de uma boneca mecânica são controlados por um circuito que consiste em ***dispositivos****. Os dispositivos são conectados por* ***tubos****. Cada dispositivo tem uma ou duas* ***saídas*** *e pode ter arbitrariamente muitas (possivelmente zero)* ***entradas****. Cada tubo conecta uma saída de um dispositivo a uma entrada do mesmo ou de outro dispositivo. Exatamente um tubo é conectado a cada entrada e exatamente um tubo é conectado a cada saída.*

Para descrever como a boneca realiza os movimentos, considere uma ***bola*** *que é colocada em um dos dispositivos. A bola viaja através do circuito. Em cada passo da viagem, a bola deixa o dispositivo usando uma de suas saídas, viaja ao longo do tubo conectado à saída e entra no dispositivo na outra extremidade do tubo.*

Existem três tipos de dispositivos: ***origem****,* ***gatilho*** *e* ***interruptor****. Existem exatamente uma origem,* M *gatilhos e* S *interruptores (*S *pode ser zero). Você deve decidir o valor de* S*.*

Cada dispositivo tem um número de série único. A origem é o dispositivo onde a bola é colocada inicialmente. Ela tem uma saída. Seu número de série é $\mathbf{0}$*.*

Um gatilho faz com que a boneca realize um movimento específico sempre que a bola entra nele. Todo gatilho tem uma saída. Os números de série dos gatilhos são de $\mathbf{1}$ *até* $\mathbf{M}$*.*

Cada interruptor tem duas saídas, que são chamadas de 'X' e 'Y'. O ***estado*** *de um interruptor é 'X' ou 'Y'. Depois que a bola entra em um interruptor, ela deixa o interruptor usando a saída dada pelo estado atual do interruptor. Depois disso, o interruptor muda seu*

estado para o oposto. Inicialmente, o estado de todo interruptor é 'X'. Os números de série dos interruptores são de $-\mathbf{1}$ *até* $-\mathbf{S}$.

Você recebe o número de gatilhos M. *Você também recebe uma sequência* A *de comprimento* N, *onde cada elemento é um número de série de um gatilho. Cada gatilho pode aparecer algumas (possivelmente zero) vezes na sequência* A. *Sua tarefa é criar um circuito que satisfaça as seguintes condições:*

- *A bola retorna à origem após alguns passos.*
- *Quando a bola retorna à origem pela primeira vez, o estado de cada interruptor é 'X'.*
- *A bola retorna à origem pela primeira vez após entrar nos gatilhos exatamente* N *vezes. Os números de série dos gatilhos, na ordem em que são entrados, são* $A_0, A_1, \ldots, A_{N-1}$.
- *Seja* P *o número total de mudanças de estado de todos os interruptores causadas pela bola antes de ela retornar à origem pela primeira vez. O valor de* P *não deve exceder* $\mathbf{20\,000\,000}$.

Ao mesmo tempo, você não quer usar muitos interruptores.

Problema 15 (IOI 2020 - Connecting Supertrees)**.** *Gardens by the Bay é um grande parque natural em Singapura. No parque existem* n *torres, conhecidas como superárvores. Essas torres são rotuladas de* 0 *a* $n-1$. *Gostaríamos de construir um conjunto de zero ou mais pontes. Cada ponte conecta um par de torres distintas e pode ser atravessada em qualquer direção. Não deve haver duas pontes conectando o mesmo par de torres.*

Um caminho da torre x *para a torre* y *é uma sequência de uma ou mais torres tal que:*

- *o primeiro elemento da sequência é* x,
- *o último elemento da sequência é* y,
- *todos os elementos da sequência são distintos, e*
- *cada dois elementos consecutivos (torres) na sequência são conectados por uma ponte.*

Note que, por definição, existe exatamente um caminho de uma torre para ela mesma e o número de caminhos diferentes da torre i *para a torre* j *é o mesmo que o número de caminhos diferentes da torre* j *para a torre* i.

O arquiteto-chefe encarregado do projeto deseja que as pontes sejam construídas de tal forma que, para todos $0 \leq i, j \leq n-1$, *existam exatamente* $p[i][j]$ *caminhos diferentes da torre* i *para a torre* j, *onde* $0 \leq p[i][j] \leq 3$.

Construa um conjunto de pontes que satisfaça os requisitos do arquiteto, ou determine que é impossível.

Problema 16 (IOI 2018 - Counting Mushrooms)**.** *Andrew, o especialista em cogumelos, está investigando cogumelos nativos de Singapura.*

Como parte de sua pesquisa, Andrew coletou n *cogumelos rotulados de* 0 *a* $n-1$. *Cada cogumelo é de uma de duas espécies, que são chamadas de* A *e* B.

Andrew sabe que o cogumelo 0 *pertence à espécie* A, *mas como as duas espécies parecem iguais, ele não sabe a espécie dos cogumelos de* 1 *a* $n-1$.

Felizmente, Andrew tem uma máquina em seu laboratório que pode ajudar com isso. Para usar esta máquina, deve-se colocar dois ou mais cogumelos em uma fileira dentro da máquina (em qualquer ordem) e ligá-la. Então, a máquina calcula o número de pares adjacentes de

cogumelos que são de espécies diferentes. Por exemplo, se você colocar cogumelos das espécies $[A, B, B, A]$ *(nessa ordem) na máquina, o resultado será* 2.

No entanto, como operar a máquina é muito caro, a máquina pode ser usada um número limitado de vezes. Além disso, o número total de cogumelos colocados na máquina em todos os seus usos não pode exceder 100 000. *Use esta máquina para ajudar Andrew a contar o número de cogumelos da espécie A coletados.*

Problema 17 (IOI 2022 - Prisoner Challenge)**.** *Em uma prisão, há* ***500*** *prisioneiros. Um dia, o diretor oferece a eles uma chance de se libertarem. Ele coloca dois sacos com dinheiro, saco A e saco B, em uma sala. Cada saco contém entre* **1** *e N moedas, inclusive. O número de moedas no saco A é* ***diferente*** *do número de moedas no saco B. O diretor apresenta aos prisioneiros um desafio. O objetivo dos prisioneiros é identificar o saco com menos moedas.*

A sala, além dos sacos de dinheiro, também contém uma lousa. Um único número deve ser escrito na lousa a qualquer momento. Inicialmente, o número na lousa é **0**.

Então, o diretor pede aos prisioneiros que entrem na sala, um por um. O prisioneiro que entra na sala não sabe quais ou quantos outros prisioneiros entraram na sala antes dele. Cada vez que um prisioneiro entra na sala, ele lê o número atualmente escrito na lousa. Após ler o número, ele deve escolher ou o saco A ou o saco B. O prisioneiro então ***inspeciona*** *o saco escolhido, ficando assim sabendo o número de moedas dentro dele. Então, o prisioneiro deve realizar uma das duas seguintes* ***ações****:*

- *Sobrescrever o número na lousa com um inteiro não negativo e sair da sala. Note que ele pode mudar ou manter o número atual. O desafio continua depois disso (a menos que todos os* ***500*** *prisioneiros já tenham entrado na sala).*
- *Identificar um saco como aquele que tem menos moedas. Isso encerra imediatamente o desafio.*

O diretor não pedirá a um prisioneiro que saiu da sala para entrar na sala novamente.

Os prisioneiros vencem o desafio se um deles identificar corretamente o saco com menos moedas. Eles perdem se algum deles identificar o saco incorretamente, ou se todos os ***500*** *prisioneiros tiverem entrado na sala e não tentarem identificar o saco com menos moedas.*

Antes do desafio começar, os prisioneiros se reúnem no salão da prisão e decidem uma ***estratégia*** *comum para o desafio em três passos.*

- *Eles escolhem um inteiro não negativo* x*, que é o maior número que eles podem querer escrever na lousa.*
- *Eles decidem, para qualquer número* i *escrito na lousa (*$0 \leq i \leq x$*), qual saco deve ser inspecionado por um prisioneiro que lê o número* i *da lousa ao entrar na sala.*
- *Eles decidem qual ação um prisioneiro na sala deve realizar após saber o número de moedas no saco escolhido. Especificamente, para qualquer número* i *escrito na lousa (*$0 \leq i \leq x$*) e qualquer número de moedas* j *visto no saco inspecionado (*$1 \leq j \leq N$*), eles decidem:*
  - *qual número entre* **0** *e* x *(inclusive) deve ser escrito na lousa, ou*
  - *qual saco deve ser identificado como aquele com menos moedas.*

Ao vencer o desafio, o diretor libertará os prisioneiros após cumprirem mais x *dias.*

Sua tarefa é conceber uma estratégia para os prisioneiros que garanta que eles vençam o desafio (independentemente do número de moedas no saco A e no saco B)

Problema 18 (IOI 2022 - Rarest Insects)**.** *Existem N insetos, indexados de 0 a $N-1$, correndo pela casa de Pak Blangkon. Cada inseto tem um* ***tipo****, que é um inteiro entre 0 e 10^9 inclusive. Múltiplos insetos podem ter o mesmo tipo.*

Suponha que os insetos sejam agrupados por tipo. Definimos a cardinalidade do tipo de inseto ***mais frequente*** *como o número de insetos em um grupo com o maior número de insetos. Similarmente, a cardinalidade do tipo de inseto* ***mais raro*** *é o número de insetos em um grupo com o menor número de insetos.*

Por exemplo, suponha que existam 11 insetos, cujos tipos são $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. Neste caso, a cardinalidade do tipo de inseto ***mais frequente*** *é 3. Os grupos com o maior número de insetos são o tipo 9 e o tipo 11, cada um consistindo de 3 insetos. A cardinalidade do tipo de inseto* ***mais raro*** *é 1. Os grupos com o menor número de insetos são o tipo 7, o tipo 0 e o tipo 100, cada um consistindo de 1 inseto.*

Pak Blangkon não sabe o tipo de nenhum inseto. Ele tem uma máquina com um único botão que pode fornecer algumas informações sobre os tipos dos insetos. Inicialmente, a máquina está vazia. Para usar a máquina, três tipos de operações podem ser realizados:

1. *Mover um inseto para dentro da máquina.*
2. *Mover um inseto para fora da máquina.*
3. *Pressionar o botão na máquina.*

Cada tipo de operação pode ser realizado no máximo **40 000** *vezes. Sempre que o botão é pressionado, a máquina relata a cardinalidade do tipo de inseto* ***mais frequente****, considerando apenas os insetos dentro da máquina.*

Sua tarefa é determinar a cardinalidade do tipo de inseto ***mais raro*** *entre todos os N insetos na casa de Pak Blangkon usando a máquina.*