XII Maratona UnB de Programação

Caderno de Problemas

04 de novembro de 2024

(Este caderno contém 15 problemas)

Comissão Organizadora:
Alberto Tavares Duarte Neto (UnB)
Arthur Botelho (UnB)
Bruno Ribas (UnB/FGA)
Daniel Porto (UnB)
Daniel Saad (IFB)
Edson Alves da Costa Júnior (UnB/FGA)
Eduardo Quirino (UnB)
Guilherme Novaes Ramos (UnB)
Jeremias Moreira Gomes (IDP)
João Henrique de Souza Pereira (UFU)
José Marcos da Silva Leite
Vinicius Ruela Pereira Borges (UnB)

Universidade de Brasília, Faculdade do Gama, DF

Lembretes

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos. Deve-se considerar entradas e saídas padrão;
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

Java

- Não declare `‘package’` no seu programa Java.
- Note que a conveção para o nome do arquivo fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual a letra que identifica o problema.

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Problema A – Areias da Anarquia

Limite de tempo: 3s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

No ano de 2040, Brasília virou um deserto. Não há mais água, servidores públicos e qualquer indício de sombra. Toda a sociedade colapsou e os sobreviventes lutam pelo que resta.

Maximiliano está na luta diária para conseguir sobreviver e percorre as regiões do DF na busca por recursos. Para isso, ele conta com seu Opala V8 e seu fiel companheiro canino, Rocky. Apesar de quase todas as estradas estarem arruinadas, elas são de mão dupla, e ainda é possível a partir de uma região e chegar em qualquer outra, direta ou indiretamente. Contudo, só existe, no máximo, um caminho entre qualquer par de regiões do Distrito Federal.

Maximiliano precisa estudar sua rota cuidadosamente, pois, apesar de ter a opção de procurar por gasolina ao chegar em cada região, se o seu carro ficar parado no meio do trajeto, ele pode ser roubado por outros sobreviventes. Sendo assim, Maximiliano precisa saber de antemão qual a estrada mais longa que liga quaisquer duas regiões vizinhas no seu percurso para determinar se ele pode ficar sem gasolina ou não.

Ajude Maximiliano a descobrir essa distância e elaborar o seu itinerário.

Entrada

A primeira linha da entrada possui três inteiros, n e q, separados por um espaço.

As próximas $n-1$ linhas descrevem cada, três inteiros u, v e w, separados por um espaço. Cada inteiro indica que existe uma estrada da região u para a região v com distância w.

As próximas q linhas contém, cada, um par de regiões a e b, indicando, respectivamente, a origem e o destino do percurso de Maximiliano.

Restrições:

- $2 \leq n \leq 10^5$
- $1 \leq u, v \leq n$, e $u \neq v$
- $0 \leq w \leq 10^9$
- $1 \leq q \leq 10^5$
- $1 \leq a, b \leq n$, e $a \neq b$

Saída

Para cada trajeto definido pelas regiões a e b, imprima a maior estrada presente nesse trajeto em uma linha.

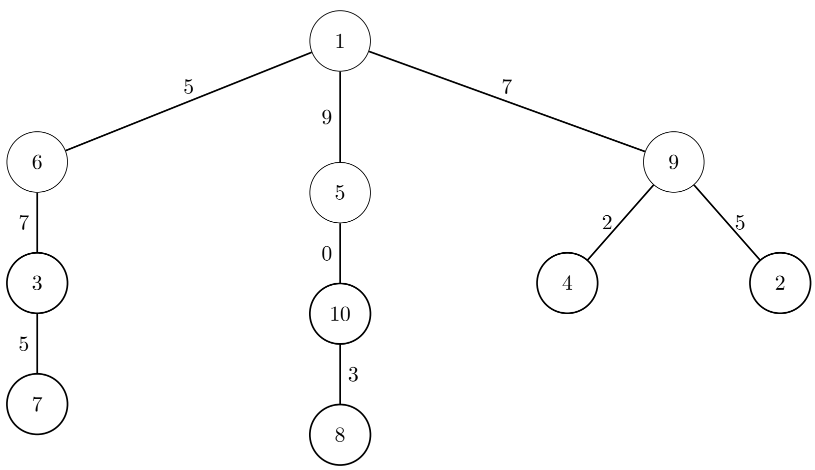
Exemplo

Entrada	Saída

`10 3`
`10 5 0`
`1 6 5`
`1 5 9`
`4 9 2`
`2 9 5`
`7 3 5`
`6 3 7`
`10 8 3`
`9 1 7`
`1 7`
`9 7`
`2 9`
	`7`
	`7`
	`5`

Notas

A figura a seguir reflete o exemplo de entrada:



No caminho da região 1 à região 7, passamos pelas regiões 6 e 3. A estrada mais longa nesse caminho possui distância 7. No caminho da região 9 até à região 7, passamos pelas regiões 1, 6 e 3. A estrada mais longa nesse caminho possui distância 7. Por fim, a região 2 é vizinha imediata da região 9 e estão ligadas por uma estrada com distância 5.

Problema B – Bravo, bravo!

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

Os artistas circenses buscam agradar o público com suas peripécias, mas as estrepolias ficam cada vez mais arriscadas para conseguir os aplausos. Não interessa à audiência as integridades física e mental deles, o espetáculo não pode parar! Entretanto, vale a pena refletir sobre esta cultura e a valorização da aparência em detrimento da essência humana.

Considerando uma apresentação circense, parabenize cada artista por sua performance inesquecível! Reflita sobre como tentar superar seus limites, por vezes a grande custo pessoal, visando reconhecimento.

Entrada

A entrada consiste de 5 linhas. As duas primeiras indicam, cada uma, uma estrepolia que o artista executa, em crescente dificuldade. Duas linhas seguintes apresentam uma nova manobra arriscada que ele tenta para agradar à platéia (e sua consequência), e são seguidas pela reação da audiência a ela.

É garantido que nenhuma linha tem mais de 100 símbolos.

Saída

Para cada estrepolia, saúde o artista com todo o entusiasmo: *"Bravo, bravo!"*. Para cada manobra arriscada/consequência, pondere sobre a cultura do espetáculo, principalmente se o custo para o artista compensa o benefício para a platéia. Depois, apresente a reação novamente.

Exemplo

Entrada	Saída

`Uma pirueta, duas piruetas`
`Superpiruetas, ultra-piruetas`
`Salta sobre a arquibancada`
`E tomba de nariz`
`Que a mocada vai pedir bis`
	`Uma pirueta, duas piruetas`
	`Bravo, bravo!`
	`Superpiruetas, ultra-piruetas`
	`Bravo, bravo!`
	`Salta sobre a arquibancada`
	`E tomba de nariz`
	`Que a mocada vai pedir bis`
	`Que a mocada vai pedir bis`

`Quatro cambalhotas, cinco cambalhotas`
`Arqui-cambalhotas, hiper-cambalhotas`
`Rompe a lona, beija as nuvens`
`Tomba de nariz`
`Que os jovens vao pedir bis`
	`Quatro cambalhotas, cinco cambalhotas`
	`Bravo, bravo!`
	`Arqui-cambalhotas, hiper-cambalhotas`
	`Bravo, bravo!`
	`Rompe a lona, beija as nuvens`
	`Tomba de nariz`
	`Que os jovens vao pedir bis`
	`Que os jovens vao pedir bis`

Notas

Atenção aos exemplos, é preciso também apresentar as informações da entrada.

Problema C – Cabo Carente

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Porto

A empresa Cabo Carente Inc., famosa por sua obsessão em cortar custos, está com um novo projeto: conectar seus dois prédios usando cabos de rede antigos que estavam jogados no depósito. Como uma empresa que preza por cada centavo, eles querem encontrar a maneira mais barata de unir todos os cabos em um único "supercabo" que ligue os prédios.

Cada cabo possui um comprimento específico. Quando dois cabos são conectados, eles se transformam em um único cabo cujo comprimento é a soma dos dois cabos conectados. O custo dessa operação é equivalente ao comprimento do maior dos dois cabos conectados. O objetivo da Cabo Carente Inc. é minimizar o custo total de todas as conexões necessárias para montar o "supercabo". Afinal, os conectores são caros!

Considerando que os cabos já foram separados, sua missão é ajudar a Cabo Carente a encontrar o menor custo para conectar todos eles e modo a conectar seus dois prédios da maneira mais econômica possível.

Entrada

A primeira linha contém um número inteiro N $(2 \leq N \leq 10^3)$, representando o número de cabos.

A segunda linha contém N inteiros C $(1 \leq C \leq 10^2)$ separados por espaços, onde cada inteiro representa o comprimento de um cabo de rede.

Saída

Um único número inteiro contendo o menor custo total para conectar todos os cabos.

Exemplo

Entrada	Saída
4 4 3 2 6	17
5 1 25 10 3 2	40

Notas

No caso do Exemplo 1: Primeiro, conectamos os cabos de comprimento 3 e 2. O custo é 3 e resulta em um novo cabo de tamanho 5. Assim temos agora cabos de comprimento [4, 6, 5].

Depois, conectamos os cabos de comprimento 4 e 5. O custo é 5 e o novo cabo possui tamanho 9. Ficamos então com cabos de comprimento [6, 9].

Finalmente, conectamos os cabos de comprimento 6 e 9. O custo é 9 e o "supercabo" possui tamanho 15.

O custo total é $3 + 5 + 9 = 17$.

Problema D – Divisores

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

Dado um inteiro positivo n, a função $\tau(n)$ retorna o número de divisores positivos de n. Por exemplo, $\tau(6) = 4$ e $\tau(11) = 2$. Dado um inteiro positivo D, determine o menor inteiro positivo n tal que $\tau(n) = D$.

Entrada

A entrada é composta por uma única linha, contendo o valor do inteiro D ($1 \leq D \leq 10^9$).

Saída

Imprima, em uma linha, o valor do menor inteiro positivo n tal que $\tau(n) = D$. Se n for maior do que 10^{18}, ao invés de n imprima o valor -1. É possível provar que, para qualquer positivo m, existe ao menos um positivo n tal que $\tau(n) = m$.

Exemplo

Entrada	Saída
6	12
15	144
314159268	-1

Notas

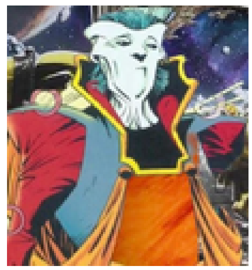
No primeiro caso, 12 tem exatamente 6 divisores positivos, a saber: $1, 2, 3, 4, 6, 12$. Observe que 18 também tem 6 divisores $(1, 2, 3, 6, 9, 18)$, porém não é uma resposta válida, uma vez que é maior do que 12.

Problema E – Estacionando na Terra

Limite de tempo: 2s
Limite de memória: 256MB

Autor: Jeremias Moreira Gomes

Boverick é um homem em uma missão, ou melhor, um alien em uma missão! Após um acidente envolvendo plástico bolha e um acelerador de partículas, ele se tornou um imortal e, pelo tédio de ter toda a eternidade pela frente, decidiu que insultará todas as personalidades famosas do universo "EM ORDEM ALFABÉTICA".



Ele encontra-se atualmente em Raxacoricofallapatorios, onde acaba de insultar Elbis Presley e está reabastecendo sua nave para se dirigir à Terra e insultar Elon Musc. Porém, neste exato momento, ele descobriu que Elon não está mais na Terra, mas ainda assim terá que ir até lá, pois o próximo a ser insultado é o Eloy Bighouse.

Dessa forma, ele decidiu que irá rastrear Elon Musc para descobrir onde ele está, irá até lá para insultá-lo e, depois, seguirá para a Terra. Boverick não sabe exatamente onde Elon está, mas tem em mãos uma lista de possíveis locais onde ele pode estar. Nessa lista, Raxacoricofallapatorios é o primeiro planeta, identificado pelo número 1, e a Terra é o último planeta, identificada pelo número N.

Agora Boverick precisa saber a quantidade de combustível necessária para cumprir sua jornada, uma vez que sempre há um caminho que leva de um planeta a qualquer outro planeta de sua lista, e que cada caminho funciona nas duas direções. Sabendo-se que Elon Musc pode estar em qualquer um dos planetas de sua lista, ele precisa calcular a menor distância, considerando o pior caso possível, saindo de Raxacoricofallapatorios, passando pelo planeta onde Elon Musc se encontra, e finalizando na Terra, a partir das distâncias entre os planetas da lista.

Para não perder tempo, mesmo tendo todo o tempo do mundo, Boverick pediu sua ajuda para calcular a menor distância que terá que percorrer em sua jornada.

Entrada

A entrada é composta por um único caso de teste. A primeira linha da entrada contém dois inteiros N ($2 \leq N \leq 4000$) e M ($1 \leq M \leq 200000$), que representam o número de planetas na lista de Boverick (de 1 a N) e o número de caminhos entre os planetas. As próximas M linhas contém três inteiros A, B e d ($1 \leq A, B \leq N$, $1 \leq d \leq 1000$) cada uma, indicando que existe um caminho de distância d entre os planetas A e B.

Saída

A saída deve conter um único inteiro, indicando a menor distância que Boverick terá que percorrer a partir de Raxacoricofallapatorios, representado pelo número 1, até a Terra, representada por N, passando antes por onde Elon Musc se encontra.

Exemplo

Entrada	Saída
5 5 3 1 2 5 2 5 2 4 2 1 5 5 4 1 5	12
5 7 5 3 3 2 3 1 2 4 1 1 3 2 1 4 2 4 3 2 1 5 3	7

Notas

No primeiro caso de testes, o pior lugar para Elon estar seria em 4, pois o caminho de Raxacoricofallapatorios até Elon e depois até a Terra seria: $1 \to 4 \to 2 \to 5$

Problema F – Fitoplâncton

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

O professor Borges está estudando o comportamento do fitoplâncton – micro-organismos fotossintetizantes que vivem flutuando na superfície das águas – no litoral brasileiro. A cada experimento de campo, ele codificou suas observações por meio de um caractere alfabético minúsculo, formando uma string S de N caracteres.

Ao analisar o comportamento dos dados obtidos, ele notou que várias destas strings apresentavam um comportamento cíclico. De modo a evidenciar tal comportamento, ele resolveu recodificar seus dados: a string S deveria ser representada por um inteiro M e uma string R tais que S fosse igual a concatenação de M cópias de R. Por exemplo, para $S =$ `"abababab"`, uma representação possível seria $M = 4$ e $R =$ `"ab"`.

Contudo, algumas strings são demasiadamente grandes para uma codificação manual e o professor Borges, embora exímio programador, tem várias novas viagens de campo agendadas. Auxilie-o escrevendo um programa que, dada uma string S, gere sua codificação. Se houver mais de uma codificação possível, retorne a que minimiza o tamanho de R.

Entrada

A entrada é composta de uma única linha, contendo uma string S ($1 \leq |S| \leq 2 \times 10^6$), composta apenas por caracteres alfabéticos minúsculos.

Saída

A saída é composta de duas linhas. A primeira delas contém o valor do inteiro M e a segunda o valor da string R, tais que ambos sejam a codificação de S que minimiza o tamanho de R.

Exemplo

Entrada	Saída
`abababab`	`4` `ab`
`rssrs`	`1` `rssrs`
`yzzyzz`	`2` `yzz`

Notas

No primeiro caso, `"abababab"= "ab"+ "ab"+ "ab"+ "ab"`. Observe que $M = 2$ e $R =$ `"abab"` também codificariam S, mas nesta R não é mínimo.

Problema G – g3r3mias

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

"Não tinham medo os tais maratonistas" era o que todos diziam quando o evento apareceu. Mas acontece que um tal de `g3r3mias`, **n**utella de renome, apareceu por lá, e ficou sabendo dos planos da tal galera e decidiu que essa festa ele ia estragar.

Apesar de ser capaz de competir (e talvez vencer) a competição, `g3r3mias` preferiu se desafiar hackeando o sistema de avaliação BOCA (*Bruno Oracula o Código Alheio*). Ele analisará todas as submissões e "incluirá" sua participação, garantindo a vitória.

A *XII Maratona UnB de Programação* segue os moldes do ICPC, isto é: ganha quem tiver mais acertos e, em caso de empate, menor tempo acumulado. Desumilde, `g3r3mias` sempre quer acertar mais problemas que os outros. Entretanto, tentando camuflar o pouco decoro de sua abordagem, evita "vencer" por mais de um problema ou por mais de um minuto.

Entrada

A entrada apresenta uma linha com dois inteiros, separados por espaço, N $(1 \leq N \leq 1000)$ e P $(1 \leq P \leq 20)$, que representam respectivamente o número de competidores, além de `g3r3mias`, e o número de problemas da prova. A seguir são apresentadas N linhas, cada uma com o resultado do i-ésimo competidor: os inteiros B_i $(0 \leq B_i \leq P)$ e T_i $(B_i \cdot 1 \leq T_i \leq B_i \cdot 300)$, separados por espaço, indicando o número de balões adquiridos e o tempo acumulado.

Saída

Apresente uma linha com dois inteiros, separados por espaço, indicando o resultado mínimo que `g3r3mias` precisa apresentar para vencer.

Exemplo

Entrada	Saída
4 10 5 255 3 89 5 488 0 0	6 255
5 12 11 255 12 555 12 666 9 488 8 25	12 554

Notas

No primeiro exemplo, `g3r3mias` "vence" com um problema a mais que o primeiro concorrente. No segundo exemplo, ele "é mais rápido" que o segundo competidor.

oracular: (verbo intransitivo) Falar como oráculo.

Problema H – Homem que Ordenava

Limite de tempo: 1s
Limite de memória: 256MB

Autor: João Henrique Souza Pereira

Era conhecido por UnB, o Único Beremias do reino de Malba. Após ajudar um vizir a realizar várias tarefas, por sua excelência, rapidez e precisão, este notável homem que ordenava foi convocado para ajudar o sultão de Malba em suas atividades de administração do reino.

O governante máximo queria saber várias informações para ajudá-lo a ser um líder melhor para o seu povo e outras nações. Um dos seus planos era ser o pioneiro em ações de sustentabilidade, dentre todos os reinos conhecidos. Para isto, ele precisava acompanhar a quantidade de árvores e seu crescimento, por tamanho, para demonstrar a evolução, a cada ano. Isso ajudaria a mostrar que suas ações tinham impacto benéfico sobre o meio ambiente, que já era conhecido como um grande reduto de Sequoias Hyperion Gigantes, com até 115 metros de altura.

Então, o sultão designou a primeira tarefa para Único Beremias: calcular a quantidade de árvores existentes, por altura. O homem que ordenava demonstrou sua grande habilidade pois, além de calcular a quantidade de árvores de cada tamanho, entregou e informação ordenada pela altura.

Entrada

A entrada possui uma sequência com uma quantidade S ($1 \leq S \leq 10^6$) de números inteiros. Cada número inteiro tem um valor N_i ($0 \leq N_i \leq 115$) correspondente ao tamanho, em metros, da i-ésima árvore do reino de Malba.

Saída

A saída mostra a quantidade de árvores existentes no reino de Malba, por tamanho em metros, ordenada pelo tamanho das árvores.

Exemplo

Entrada	Saída
1 2 3 4	1 1 2 1 3 1 4 1
108 74 89 82 46 89 80 89 74	46 1 74 2 80 1 82 1 89 3 108 1
37 78 37 37 20 78	20 1 37 3 78 2

Problema I – Império Quadradônico

Limite de tempo: 3s
Limite de memória: 256MB

Autor: José Leite

A cidade de Nlogônia foi erguida na mesma região onde, séculos atrás, existia a antiga cidade de Quadradônia.

Recentemente, foi descoberto um importante documento arqueológico contendo mapas de Quadradônia, que foi traduzido para nosso sistema de coordenadas. Agora, a população se pergunta quanto da área de Nlogônia pode ser alvo de futuras escavações, isto é, o quanto de sua área foi construída sobre as ruínas de Quadradônia. O mapa da cidade de Nlogônia é bem conhecido.

Os contornos de cada cidade podem ser vistos como polígonos simples, sem buracos ou interseções em lados não consecutivos, embora possam ser não convexos. Como você é o próximo mestre em geometria, a cidade conta com sua ajuda para calcular essa área de intercessão!

Dados os polígonos correspondentes as cidades Nlogônia e Quadradônia, calcule a área da nova cidade que sobrepõe Quadradônia.

Entrada

A primeira linha da entrada contem um inteiro $t (1 \leq t \leq 500)$, o número de casos de teste.

A primeira linha de cada caso de teste contém um inteiro $N (1 \leq N \leq 1000)$.

Cada uma das próximas N linhas contem dois inteiros $x_i, y_i (1 \leq x_i, y_i \leq 2000)$, separados por espaço, indicando as coordenadas de Nlogônia. Os pontos serão dados em ordem anti-horária.

A próxima linha contém um inteiro $M (1 \leq M \leq 1000)$.

Cada uma das próximas M linhas contem dois inteiros $x_i, y_i (1 \leq x_i, y_i \leq 2000)$, separados por espaço, indicando as coordenadas da cidade antiga de Quadradônia. Os pontos serão dados em ordem anti-horária.

É garantido que a soma de N, para todos os casos de teste, não excede 2000.

É garantido que a soma de M, para todos os casos de teste, não excede 2000.

Saída

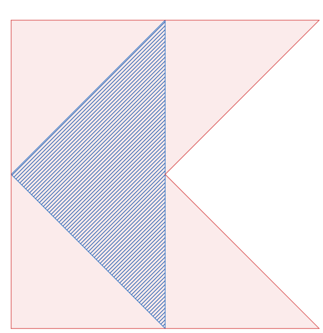
Para cada caso de teste, imprima a área de intercessão. A resposta será aceita caso o erro absoluto ou relativo seja menor ou igual a 10^{-6}. Mais formalmente, seja a a sua resposta e b a área de interseção, então a sua resposta será considerada correta caso $\frac{|a-b|}{max(1,b)} < 10^{-6}$.

Exemplo

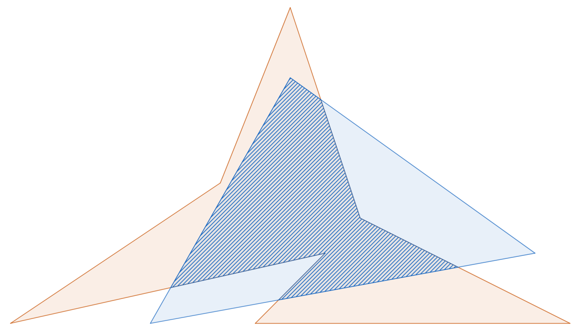
Entrada	Saída
3 3 1 2 2 1 2 3 5 2 2 3 3 1 3 1 1 3 1 7 3 1 12 3 10 1 19 1 13 4 11 10 9 5 3 7 1 18 3 11 8 3 1 2 3 1 3 4 5 2 3 4 4 1 4 1 2 4 2	1.00000000000000000000 20.56136363636363636326 1.50000000000000000000

Notas

No primeiro caso de teste, Nlogonia está totalmente dentro da região de Quadradonia.



O segundo caso de teste é representado por:



Problema J – Jotinha e string

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Alberto Neto

Todos sabem que Jotinha adora strings. Em seu aniversário, Alexandre o presenteia com uma string s de tamanho n. Jotinha percebe que cada substring t de s tem um valor dado por $f(t) \cdot |t|$, onde $f(t)$ é a quantidade de substrings de s iguais a t, e $|t|$ é o tamanho de t.

Jotinha, então, decide coletar todas as substrings de s em um vetor e ordená-las pelo seu valor. Se duas strings possuem o mesmo valor, a menor lexicograficamente é colocada primeiro. Dado um inteiro k, Jotinha quer encontrar a k-ésima string do vetor ordenado.

Ajude o estudante da PUC e escreva um programa para encontrar a k-ésima string!

Entrada

A primeira linha de entrada contém um único inteiro n ($1 \leq n \leq 2 \times 10^5$) – o tamanho da string s que Jotinha recebe.

A segunda linha contém uma cadeia de n caracteres alfabéticos minúsculos – a string s ($|s| = n$).

A terceira linha contém um único inteiro k ($1 \leq k \leq \frac{n(n+1)}{2}$) – o valor k representando a k-ésima string desejada.

Saída

Imprima uma única linha contendo uma cadeia de caracteres – a k-ésima string do vetor ordenado de substrings de s.

Exemplo

Entrada	Saída
6 banana 4	n
9 aaaaaaaaa 45	aaaaa

Notas

As primeiras 12 substrings ordenadas, sendo o critério apresentado, da string banana são: “b”, “ba”, “n”, “n”, “a”, “a”, “a”, “ban”, “nan”, “an”, “an”.

Problema K – Kátia e as árvores

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

Kátia estava finalizando seu exercício da disciplina Estruturas de Dados, que consistia em gerar um certo número de árvores diferentes, quando notou seu erro: ela estava gerando grafos conectados, não-direcionados, com exatamente N vértices e N arestas, ou seja, estes grafos não eram árvores!

Embora ela consiga corrigir cada grafo rapidamente, o volume de trabalho é muito grande e o professor John marcou a entrega para amanhã! Auxilie Kátia escrevendo um programa que recebe um grafo conectado, não-direcionado, com exatamente N vértices e N arestas, e indique qual aresta deve ser removida para que o grafo resultante seja uma árvore.

Entrada

A primeira linha da entrada contém o valor do inteiro N ($3 \leq N \leq 2 \times 10^5$).

As N linhas seguintes contém, cada uma, um par de vértices u e v ($1 \leq u, v \leq N, u \neq v$), separados por um espaço em branco, indicando que há uma aresta bidirecional entre os vértices u e v. É garantido que o grafo resultante é conectado e simples.

Saída

Imprima, em uma linha, o identificador da aresta a ser removida para que o grafo resultante seja uma árvore. As arestas são identificadas por um número inteiro de 1 a N, de acordo com a ordem da entrada.

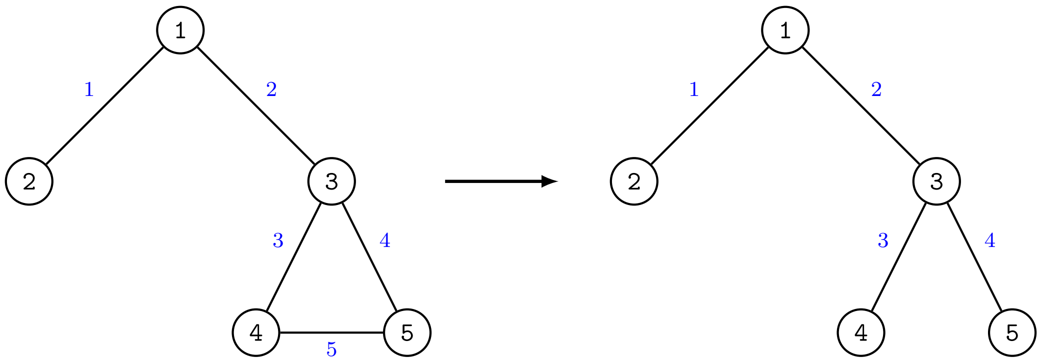
Se mais de uma aresta pode ser removida para gerar uma árvore, imprima qualquer uma delas.

Exemplo

Entrada	Saída
5 1 2 1 3 3 4 3 5 4 5	5
3 3 2 1 3 2 1	1

Notas

No primeiro caso, a remoção da aresta 5 (que conecta os vértices 4 e 5) resulta em uma árvore. Veja a figura abaixo, onde o rótulo das arestas (em azul) são os seus identificadores:


1
2
3
4
5
1
2
3
4
5
1
2
3
4
5
1
2
3
4
5
1
2
3
4

Problema L – Logística

Limite de tempo: 7s
Limite de memória: 1024MB

Autor: Arthur Botelho

Uma empresa de logística possui um galpão retangular representado por uma matriz de n linhas e m colunas. Cada uma das células da matriz representa um pacote com $a_{i,j}$ $(1 \leq i \leq n, 1 \leq j \leq m)$ itens a serem transportados.

Para transportar os pacotes, seus itens são divididos em caixas. Por motivos de segurança, nunca pode-se colocar itens de pacotes diferentes em uma mesma caixa, nem deixar de preencher completamente alguma caixa usada no transporte (se uma caixa possui tamanho k, ela é usada para transportar k itens de um mesmo pacote de uma vez, necessariamente).

Essa empresa possui um modo bastante peculiar de operar: ela transporta de uma vez todos os pacotes localizados em uma submatriz da matriz do galpão. Para fazer isso, ela deve previamente escolher um inteiro positivo k que será o tamanho de todas as caixas usadas no transporte desses pacotes. Além de escolher de forma a respeitar as restrições sobre transporte de itens nas caixas, o k será escolhido de forma a minimizar a quantidade total de caixas usadas no transporte desse conjunto de pacotes. Assim, define-se o custo mínimo de transporte da região $((A, B), (C, D))$ como a menor quantidade possível de caixas de um mesmo tamanho que podem ser usadas para transportar todos os itens dos pacotes nas posições (i, j) tais que $A \leq i \leq B$ e $C \leq j \leq D$.

Enquanto aguarda por pedidos de transporte, a empresa decidiu realizar uma estimativa usando duas listas crescentes de inteiros x e y, tais que $1 \leq x_i \leq n$ e $1 \leq y_i \leq m$. Mais especificamente, deseja-se saber a média dos custos mínimos de transportes de todas as regiões da forma $((x_a, x_b), (y_c, y_d))$, sendo $1 \leq a \leq b \leq |x|$ e $1 \leq c \leq d \leq |y|$.

Essa média pode ser expressa por uma fração p/q. Em vez de imprimir o valor p/q, imprima $p \cdot q^{-1} \pmod{998244353}$.

Entrada

A primeira linha da entrada contém as dimensões da matriz: n e m $(1 \leq n, m \leq 1000)$.

As próximas n linhas contêm m inteiros cada: na i-ésima dessas linhas, o j-ésimo número é o valor $a_{i,j}$ $(1 \leq a_{i,j} \leq 10^9)$.

A linha seguinte da entrada contém o valor $|x|$, o tamanho da lista x $(1 \leq |x| \leq 100)$. A próxima linha contém $|x|$ inteiros crescentes x_i $(1 \leq x_i \leq n)$.

A linha seguinte da entrada contém o valor $|y|$, o tamanho da lista y $(1 \leq |y| \leq 100)$. A próxima linha contém $|y|$ inteiros crescentes y_i $(1 \leq y_i \leq m)$.

Saída

Imprima, em uma única linha, o valor $p \cdot q^{-1} \pmod{998244353}$, onde p/q é a média desejada.

Exemplo

Entrada	Saída
3 3 1 2 3 4 5 6 7 8 9 1 2 1 3	1
3 3 2 4 6 8 12 14 16 20 24 2 2 3 1 2	332748121

Problema M – Maturação de Queijos

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Vinicius Borges

Batista é um mineiro da região da Serra da Canastra que recentemente se mudou para Brasília para fundar uma queijaria, objetivando produzir e vender queijos artesanais. Cada queijo possui uma maneira específica de ser produzido e maturado, o que envolve até os momentos exatos de início e final de sua maturação. Cada queijo também possui um custo de maturação, que inclui a matéria-prima, os recursos da fábrica e a mão-de-obra.

A câmara comporta a maturação de um queijo por vez, pois Batista acredita que se dois ou mais queijos forem colocados ao mesmo tempo, o sabor deles não seguirá as tradicionais receitas da Serra da Canastra. Batista possui um catálogo de N queijos artesanais, cada um contendo a descrição dos momentos de início s_i e final f_i de sua maturação, além do custo c_i de produção.

Determine a quantidade máxima de queijos de tipos distintos que ele pode produzir, sem exceder os recursos financeiros que Batista possui à disposição.

Entrada

A primeira linha da entrada contém dois números inteiros N $(1 \leq N \leq 500)$ e K $(1 \leq K \leq 500)$, indicando a quantidade de tipos de queijo no catálogo de Batista e o custo máximo de produção que ele pode arcar, respectivamente.

Em seguida, existem N linhas, cada uma descrevendo os tipos de queijos. A i-ésima linha apresenta três números inteiros s_i, f_i e c_i $(1 \leq s_i < f_i \leq 10^4, 1 \leq c_i \leq 10^3)$ representando o momento de início de início da maturação, o momento final da maturação, e o custo de produção do i-ésimo queijo, respectivamente.

Saída

Imprima um único número inteiro indicando a quantidade máxima de tipos de queijos que Batista pode produzir sem exceder os recursos financeiros que ele possui.

Exemplo

Entrada	Saída
3 5 1 2 1 2 3 2 3 4 2	3
4 9 1 4 11 2 5 6 8 10 2 4 7 3	2
7 17 2 3 2 1 4 5 3 4 7 5 10 4 1 10 3 2 10 5 12 14 3	4

Notas

No primeiro exemplo, pode-se maturar três queijos: o Queijo 1 pode ser maturado entre os instantes 1 e 2; o queijo 2 pode ser maturado logo após o queijo 1, entre os instantes 2 e 3; por fim, o o queijo 3 pode ser maturado após o queijo 2. O custo de produção dos três queijos não excedem os recursos financeiros de Batista.

No segundo exemplo, temos algumas combinações que maximizam a quantidade de queijos que Batista pode produzir dentro do seus recursos financeiros. Uma opção é escolher o queijo 2 e o queijo 3, ou o queijo 4 e o queijo 3, resultando em um total de 2 queijos. Portanto, a quantidade máxima de queijos que ele pode produzir é 2.

No terceiro exemplo, a combinação dos queijos que permitem a existência de um queijo maturando por vez na câmara e que não excedem o custo máximo é os queijos 1, 3, 4, e 7, totalizando um custo igual a 16.

Problema N – Nave Extraterrestre

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Vinicius Borges

Após avistarem supostos objetos voadores não identificados (OVNIs), a população de Nlogônia enviou várias imagens para um jornal local em busca de uma explicação. O jornal, por sua vez, entrou em contato com renomados ufólogos, que analisaram as imagens recebidas e recortaram as partes correspondentes à pintura dos OVNIs.

Utilizando seus conhecimentos, os ufólogos identificaram um padrão de pintura nas naves que diferencia as de origem extraterrestre das demais. No caso das naves extraterrestres, a pintura aparenta seguir um padrão de tabuleiro de xadrez, isto é, uma malha retangular homogênea onde cada célula tem uma cor distinta das células vizinhas, tanto na vertical quanto na horizontal.

A população, aterrorizada, aguarda por novas informações, temendo que uma invasão alienígena esteja prestes a começar. Ajude os ufólogos a determinar se o padrão de pintura em uma imagem digital específica está associado a uma nave extraterrestre ou não.

Entrada

A primeira linha da entrada contém dois números inteiros separados por espaço N e M $(2 \leq N, M \leq 100)$ indicando as dimensões da imagem digital contendo o padrão de pintura do OVNI.

Em seguida, existem N linhas descrevendo a pintura do OVNI. Cada linha i contém M caracteres $a_{i,1}, \ldots, a_{i,M}$ $(a_{i,j} \in \{\text{`}\mathtt{0}', \text{`}\mathtt{1}'\})$ indicando a cor do $i-$ésimo quadrado, que pode ser preta (valor zero) ou branca (valor 1).

Saída

Imprima "`S`" (sem aspas duplas) caso o padrão de pintura da imagem digital do OVNI esteja associada com uma nave alienígena ou "`N`" (sem aspas duplas) caso contrário.

Exemplo

Entrada	Saída

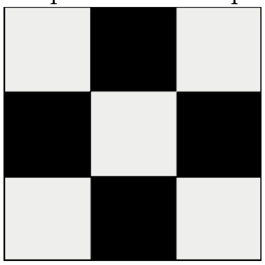
`3 3`
`101`
`010`
`101`
	`S`

`4 2`
`00`
`01`
`11`
`01`
	`N`

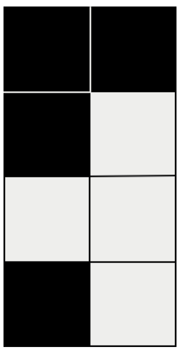
`3 5`
`01010`
`10101`
`01010`
	`S`

Notas

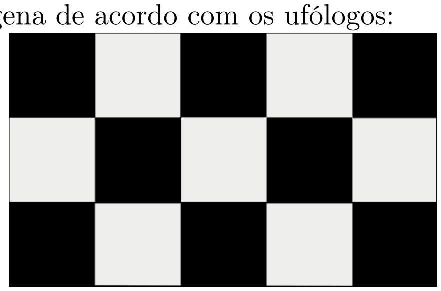
No primeiro exemplo, o seguinte padrão de pintura é fornecido aos ufólogos.



No segundo exemplo, associado à figura abaixo, algumas células no padrão de pintura não apresentam vizinhos de cores distintas na vertical e na horizontal.



No terceiro exemplo, observa-se que o padrão fornecido segue a descrição de uma nave alienígena de acordo com os ufólogos:



Problema O – O Jogo Aleatório

Limite de tempo: 5s
Limite de memória: 256MB

Autor: Arthur Botelho

Dhara e Arthur estavam sem nada para fazer no final de semana, então decidiram jogar n partidas de "O Jogo Aleatório". Esse jogo não é competitivo (não existe vencedor), e as escolhas dentro dele são feitas de forma completamente aleatória. A graça do jogo é observar como fica a disposição das peças e do tabuleiro usado após o fim de cada partida.

Para jogar uma partida, escolhe-se previamente o tabuleiro no qual vai ser jogada. Há infinitos tabuleiros para serem escolhidos, com todos os tamanhos inteiros positivos possíveis $(1, 2, 3, 4, 5...)$. Se o tabuleiro escolhido tem tamanho k, então ele possui k posições numeradas de 1 a k. Denotaremos o tabuleiro de tamanho k por $T(k)$, e por $T(k)_i$ a posição de número i do tabuleiro $T(k)$ $(1 \leq i \leq k)$.

O tabuleiro $T(k)$ vem com k peças numeradas p_i $(1 \leq i \leq k)$. As peças são fabricadas em forma de disco, de forma que a peça p_i tem o número i escrito em uma das faces, e nada escrito na outra. As peças são indistinguíveis, exceto pelo número escrito.

Ao começar uma partida, cada posição $T(k)_i$ inicia contendo a peça p_i — equivalente a dizer que a peça p_i inicia o jogo posicionada em $T(k)_i$. As posições $T(k)_i$ têm espaço suficiente para conter qualquer número de peças, mas uma peça p_i só pode estar posicionada em no máximo em uma posição $T(k)_j$ de cada vez. Dizemos que uma peça está ativa se ela está no tabuleiro e se sua face com o número escrito está virada para cima. Caso contrário, ela é dita inativa. Todas as peças começam ativas, ou seja, com o número escrito virado para cima. Arthur controla as peças com números ímpares, e Dhara, as com números pares.

Toda partida segue regras rígidas. A partir do início do jogo, em que em cada posição $T(k)_i$ inicia contendo a peça ativa p_i, jogam-se rodadas enquanto existirem peças ativas no tabuleiro $T(k)$ (ou seja, enquanto existir qualquer i tal que $T(k)_i$ contém alguma peça). Em cada rodada, escolhe-se aleatoriamente o jogador da vez (Arthur ou Dhara), a não ser que um dos jogadores não possua peças ativas sob seu controle. Neste caso, a vez é do outro jogador.

O jogador da vez escolhe (aleatoriamente), entre as peças que controla, alguma que ainda esteja ativa. Depois, decide (aleatoriamente) se a peça irá se mover ou se tornar inativa. Quando uma peça p_i deve se tornar inativa, vira-se para cima a sua face sem nada escrito, sem mudar sua posição (se p_i estava posicionada em $T(k)_j$, continua em $T(k)_j$). Uma peça p_i que irá se mover, se estava posicionada em $T(k)_j$, é movida para a posição $T(k)_{j-1}$ ($T(k)_{j-1}$ passa a conter p_i e $T(k)_j$ deixa de contê-la) e continua ativa, exceto se $j = 1$. Se isto ocorrer, a peça é retirada do tabuleiro (de modo que nenhuma posição $T(k)_j$ contém essa peça, e essa peça não está posicionada em nenhuma posição $T(k)_j$), sendo considerada inativa.

Depois do fim da partida, o tabuleiro $T(k)$ fica de tal forma que todas as suas posições $T(k)_j$ contêm um número não negativo de peças inativas (viradas para baixo), e a quantidade total de peças que terminam no tabuleiro é possivelmente menor do que k. Note que um mesmo tabuleiro pode terminar uma partida disposto de diferentes formas, devido ao caráter aleatório do jogo. Se duas partidas P_1 e P_2 são jogadas em $T(k)$, é possível que a disposição do tabuleiro após o fim de P_1 seja diferente da disposição após P_2, ou seja, pode existir alguma posição i tal que a quantidade de peças que terminaram posicionadas em $T(k)_i$ após P_1 é diferente da quantidade das que terminaram em $T(k)_i$ após P_2.

Arthur e Dhara jogaram as suas n partidas com base em uma sequência A (de tamanho n). Isso significa que, na partida i, foi escolhido o tabuleiro de tamanho A_i $(1 \leq i \leq n)$. Dhara, entre os dois, é quem mais gosta de guardar recordações em fotos. Por isso, depois do término de cada partida i, ela tira uma foto de como ficou o tabuleiro $T(A_i)$ ao final desse jogo (sem remover as peças). Depois de jogadas as n partidas, ela acabou formando uma sequência S de n fotos.

Uma foto S_i é apenas um registro do tabuleiro usado na partida i após o seu fim: se nela usamos o tabuleiro $T(A_i)$, a foto S_i desse tabuleiro após o término do jogo é equivalente a uma lista $L(S_i)$ com A_i posições numeradas de 1 a A_i, em que $L(S_i)_j$ ($1 \leq j \leq A_i$) representa o número de peças inativas que terminaram o jogo posicionadas em $T(A_i)_j$. Note que ao final de cada partida só restam no tabuleiro peças inativas, que são indistinguíveis entre si por estarem com a face com número escrito virada para baixo. Assim, dizemos que duas fotos f_1 e f_2 são iguais se $L(f_1) = L(f_2)$, ou seja, se equivalem à listas iguais (duas listas são iguais se possuem o mesmo tamanho e seus elementos em cada posição são iguais).

Observando a sequência S, Arthur percebeu que as fotos eram distintas par a par, ou seja, não existiam quaisquer duas posições distintas i, j tais que $S_i = S_j$. Dhara então se perguntou se isso poderia acontecer se, em vez de terem jogado as partidas com base na sequência A, tivessem escolhido usando uma outra sequência B (também de tamanho n).

Se fizessem isso, ao final de n partidas, eles também teriam uma sequência de fotos, S', de tamanho n. Cada uma dessas partidas pode terminar com diferentes configurações do tabuleiro (não importa que algumas configurações sejam mais prováveis que outras), resultando que cada partida i pode gerar diferentes possibilidades de foto para S_i. Ou seja, após jogar partidas com base em B, existem diferentes sequências de fotos S' possíveis de serem formadas ao final dos n jogos.

Dada a sequência B, Dhara deseja saber quantas sequências de fotos S', dentre as possíveis de serem formadas após n jogos com base em B, são tais que não existem i, j distintos entre 1 e n tais que $S'_i = S'_j$, ou seja, que as fotos de S' são distintas par a par. Imprima essa quantidade mod $10^9 + 7$.

Entrada

A primeira linha da entrada contém um inteiro n, a quantidade de partidas a serem jogadas ($1 \leq n \leq 10^5$).

A segunda linha da entrada contém n inteiros B_i representando B: a sequência de tamanhos dos tabuleiros a serem escolhidos para cada partida ($1 \leq B_i \leq 10^6$).

Saída

Imprima o número de possíveis sequências diferentes de fotos S' tais que S'_i são distintas par a par e S'_i registra uma possível disposição pós-jogo de um tabuleiro $T(B_i)$, mod $10^9 + 7$.

Exemplo

Entrada	Saída
3 2 1 2	40
3 1 1 1	0

Notas

Usando o tabuleiro $T(1)$, Arthur é o único que pode jogar e só tem uma peça para escolher. Então, necessariamente, essa peça ou irá se tornar inativa e permanecer na posição $T(1)_1$ ou será removida do tabuleiro (também se tornando inativa). Depois disso, não haverá peças ativas em jogo, e a partida termina. Assim, há 2 possibilidades para uma foto desse tabuleiro após o final de um jogo. Portanto, no primeiro caso de teste, há duas possibilidades para a foto S_2', e no segundo caso de teste não é possível tirar três fotos distintas, já que elas só podem ser de duas formas.

Usando $T(2)$, existem 5 possibilidades para o tabuleiro ao final da partida:

1. Nenhuma peça no tabuleiro.
2. Apenas uma peça restando, na posição $T(2)_1$.
3. As duas peças na posição $T(2)_1$.
4. Apenas uma peça restando, na posição $T(2)_2$.
5. Uma peça em $T(2)_1$ e outra em $T(2)_2$, como no início do jogo.