

Competidor(a): _____

Número de inscrição: _____ - _____ (opcional)

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (13 de setembro de 2024).



Olimpíada Brasileira de Informática
Competição Feminina - OBI2024

Caderno de Tarefas
Modalidade Programação • Nível 2 • Fase Única

13 de setembro de 2024

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 13 páginas (não contando a folha de rosto), numeradas de 1 a 13. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln, read, writeln, write*;
 - em C: *scanf, getchar, printf, putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
 - em Python: *read, readline, readlines, input, print, write*
 - em Javascript: *scanf, printf*
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Bibi e a árvore

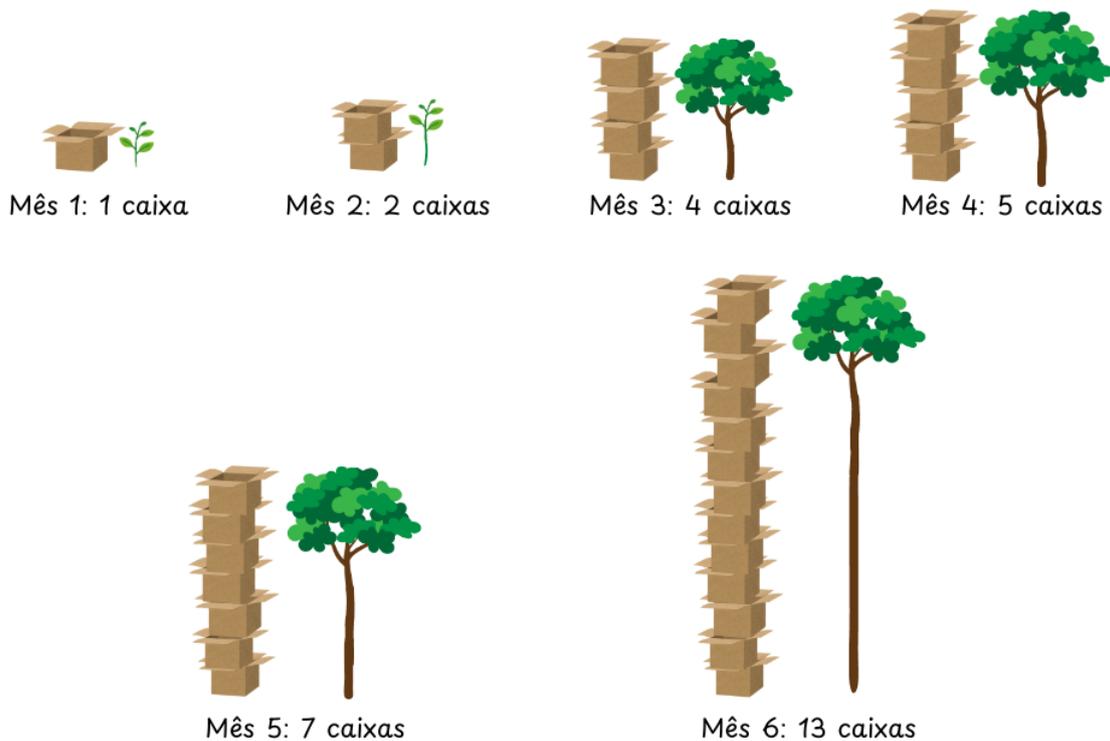
Nome do arquivo: `bibi.c`, `bibi.cpp`, `bibi.java`, `bibi.js` ou `bibi.py`

Bibi é uma menina apaixonada por natureza. Influenciada por sua mãe, que possui um lindo jardim botânico, ela começou desde cedo a plantar e cuidar de diversos tipos de plantas.

Bibi é extremamente curiosa e sonha em um dia ser uma grande cientista. Por isso, ela registra e acompanha, de forma independente, o crescimento de todas as plantas do jardim, anotando tudo em um livro. Ela também é bastante engenhosa com as ferramentas que possui à sua disposição: como ela não possui uma fita métrica, ela mede a altura das plantas usando caixinhas de papelão que estão prestes a ir para a reciclagem.

Num belo dia, sua mãe lhe trouxe de presente uma semente de *Abratibum*, uma árvore que supostamente vive até 100.000 anos (1.200.000 meses) e que seria a de maior altura já registrada no livro de Bibi. *Uau!*

Bibi começou imediatamente a registrar o crescimento mensal da árvore. Nos 6 primeiros meses, ela obteve os seguintes resultados:



Depois do sexto mês, Bibi observou que o crescimento da árvore parecia ter ficado fixo, e logo ela imaginou que isso poderia ser verdade para todos os meses seguintes. Deste modo, ela fez a seguinte anotação em seu caderno:

Quando a Abratibum chega aos 5 meses de vida, ela atinge a vida adulta. Por isso, a partir de seu sexto mês de vida, a cada mês a Abratibum crescerá em altura a mesma quantidade de caixinhas que cresceu entre o seu quinto e o seu sexto mês de vida.

Bibi notou que não terá caixinhas o suficiente para medir a Abratibum por muitos meses. Por isso, decidiu começar a registrar previsões de crescimento da árvore.

Sua tarefa é, dado um inteiro X , determinar a altura da árvore (em quantidade de caixinhas de papelão) no X -ésimo mês de vida dela.

Entrada

A entrada contém um único inteiro X , o mês de vida da Abratibum a ser consultado.

Saída

Seu programa deve produzir uma única linha contendo um único inteiro, a altura (em caixinhas de papelão) da Abratibum no seu X -ésimo mês de vida.

Restrições

- $1 \leq X \leq 1\,200\,000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta sub tarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (23 pontos):** $X \leq 7$.
- **Subtarefa 3 (27 pontos):** $5 \leq X \leq 12\,000$.
- **Subtarefa 4 (31 pontos):** $5 \leq X \leq 120\,000$.
- **Subtarefa 5 (19 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 1	Exemplo de saída 1 1
Exemplo de entrada 2 5	Exemplo de saída 2 7

Mistura de Poções

Nome do arquivo: `mistura.c`, `mistura.cpp`, `mistura.java`, `mistura.js` ou `mistura.py`

Lulu é uma estudante de magia que possui uma prateleira com N poções mágicas dispostas lado-a-lado. Cada poção tem um tipo específico, sendo a_i o tipo da i -ésima poção da esquerda para a direita.

Com a chegada de seus novos lagartos de estimação, Lulu precisa liberar espaço na prateleira. Para isso, ela pretende remover uma certa quantidade de poções do canto esquerdo e uma certa quantidade de poções do canto direito da prateleira, mantendo no final um segmento contíguo de poções na prateleira.

Além disso, Lulu quer garantir que seja possível criar pelo menos um feitiço usando as poções restantes na prateleira. Para criar um feitiço, ela precisa misturar K poções de tipos diferentes, sendo K um número pequeno.

Lulu percebeu que podem existir muitos modos de liberar espaço da forma como ela deseja. Indecisa sobre como fazer isso, ela pediu para você escrever um programa que determine a quantidade de formas diferentes de remover poções dos cantos da prateleira, atendendo às suas restrições.

Entrada

A primeira linha da entrada contém dois inteiros N e K , representando a quantidade de poções na prateleira e o número de poções de tipos diferentes necessárias para criar um feitiço, respectivamente.

A segunda linha da entrada contém N inteiros a_1, a_2, \dots, a_N representando os tipos de cada poção na prateleira, da esquerda para a direita.

Saída

Seu programa deverá imprimir uma única linha contendo um único inteiro, a quantidade de formas de remover poções dos cantos da prateleira atendendo às restrições de Lulu.

Restrições

- $1 \leq N \leq 100\,000$
- $2 \leq K \leq 3$
- $1 \leq a_i \leq N$ para todo $1 \leq i \leq N$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (13 pontos):** $N \leq 100$.
- **Subtarefa 3 (17 pontos):** $N \leq 1\,000$.
- **Subtarefa 4 (29 pontos):** $K = 2$.
- **Subtarefa 5 (41 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 4 2 2 1 2 3	Exemplo de saída 1 6
Exemplo de entrada 2 7 3 3 1 1 5 3 7 3	Exemplo de saída 2 12
Exemplo de entrada 3 2 2 1 2	Exemplo de saída 3 1
Exemplo de entrada 4 5 3 3 2 3 3 2	Exemplo de saída 4 0

Tabuleiro

Nome do arquivo: `tabuleiro.c`, `tabuleiro.cpp`, `tabuleiro.java`, `tabuleiro.js` ou `tabuleiro.py`

Júlia gosta muito de jogos de tabuleiro. Sabendo disso, seu avô Radeck a presenteou com um jogo matemático de tabuleiro.

O jogo é jogado em um tabuleiro com dimensões $3 \times N$ (3 linhas e N colunas), inicialmente vazio, e contém uma bolsa com peças de tabuleiro nos nove formatos da figura abaixo. Existem infinitas peças de cada formato.

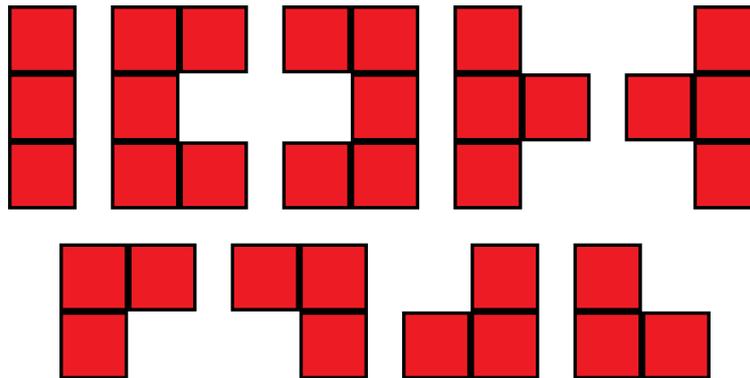


Figura 1: Peças do tabuleiro

O objetivo do jogo é determinar de quantas maneiras é possível preencher completamente o tabuleiro com peças nos formatos acima. Cada quadrado do tabuleiro deve estar coberto por exatamente uma peça. Não é permitido rotacionar as peças (ou seja, cada peça deve ser posicionada exatamente como na figura acima). Também não é permitido que sobrem pedaços de peças para fora do tabuleiro.

Por exemplo, um tabuleiro 3×2 pode ser preenchido de 3 maneiras diferentes:

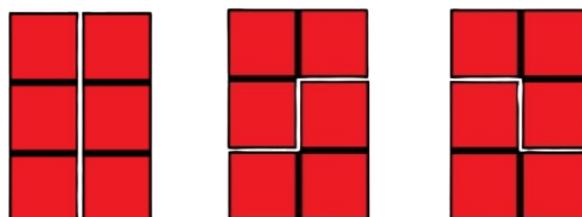


Figura 2: Possíveis soluções para um tabuleiro 3×2

Sua tarefa é, dado o comprimento N do tabuleiro, calcular o número de modos de preencher o tabuleiro de acordo com as regras acima. Se este número for maior ou igual a 1 000 000 007, você deve encontrar o resto da divisão do número por 1 000 000 007.

Entrada

A entrada consiste de uma única linha com um único inteiro N , o comprimento do tabuleiro.

Saída

Seu programa deve imprimir uma única linha contendo apenas um inteiro, o resto da divisão do número de maneiras diferentes de preencher o tabuleiro por 1 000 000 007.

Restrições

- $1 \leq N \leq 100\,000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (21 pontos):** $N \leq 20$.
- **Subtarefa 3 (42 pontos):** $N \leq 100$.
- **Subtarefa 4 (37 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 2	Exemplo de saída 1 3
----------------------------------	--------------------------------

Exemplo de entrada 2 4	Exemplo de saída 2 15
----------------------------------	---------------------------------

Exemplo de entrada 3 5	Exemplo de saída 3 35
----------------------------------	---------------------------------

Observação

O “resto da divisão” é o valor que sobra quando um número é dividido por outro. Por exemplo, o resto da divisão de 10 por 4 é 2. Este resto pode ser calculado utilizando o operador `%` entre o dividendo e o divisor. Por exemplo:

- O resto da divisão de 10 por 4 é calculado por `10 % 4`
- O resto da divisão de M por 1 000 000 007 é calculado por `M % 1000000007`

Christina e os bombons

Nome do arquivo: `bombons.c`, `bombons.cpp`, `bombons.java`, `bombons.js` ou `bombons.py`

A Competição de Programação Feminina está sendo organizada no famoso reino da Quadradolândia. Christina, que é grande fã de viagens e de chocolate, ficou responsável por levar bombons para as competidoras.

A Quadradolândia é um lugar lindo com várias montanhas e vales. O mapa do reino é descrito como um tabuleiro $L \times C$ (L linhas e C colunas), onde cada quadrado representa uma cidade do reino. As cidades da Quadradolândia possuem diferentes tipos de terreno, alguns mais fáceis de atravessar e outros mais difíceis. Por isso, cada cidade possui uma *dificuldade* p associada a ela, um inteiro expressando o quão trabalhoso é atravessar a cidade.

Christina está na cidade A , onde ela vai comprar bombons. Ela pretende andar pelo reino para levar os bombons para a competição, que acontecerá em uma outra cidade B daqui a D dias. Porém, Christina tem medo de se perder, e por isso só anda nas direções Norte, Sul, Leste e Oeste, sem sair do reino (ou seja, Christina só pode andar horizontalmente ou verticalmente dentro do tabuleiro).

Os bombons são bem pesados e, quanto mais bombons Christina carrega, mais devagar ela anda. Mais especificamente, se ela estiver levando W bombons, ela irá demorar $p \times W$ dias para atravessar uma cidade com dificuldade p . Note que Christina precisa atravessar a cidade A para sair dela, mas acaba sua viagem imediatamente ao entrar na cidade B (ou seja, ela não precisa atravessar a cidade B).

Além disso, a Quadradolândia é um reino com muitos perigos: algumas cidades possuem monstros e Christina decidiu que ela não pode passar por tais cidades. Por sorte, Christina sabe que existe pelo menos um caminho entre A e B que não atravessa cidades com monstros.

Christina quer levar o maior número possível de bombons para a competição. Ajude ela a encontrar a quantidade máxima de bombons que ela pode carregar de forma a chegar na cidade B a tempo da competição.

Entrada

A primeira linha da entrada contém três inteiros L , C e D , indicando, respectivamente, o número de linhas no tabuleiro que descreve a Quadradolândia, o número de colunas no tabuleiro e o número de dias que faltam para a competição acontecer.

As próximas L linhas contém C inteiros cada e representam a dificuldade do terreno de cada cidade. O j -ésimo inteiro da i -ésima destas linhas, p_{ij} , corresponde à dificuldade da cidade na posição (i, j) do tabuleiro. Se a cidade (i, j) possui monstros, $p_{ij} = -1$ e Christina não pode atravessá-la.

A próxima linha contém dois inteiros L_1 e C_1 , indicando que a cidade A onde Christina vai comprar os bombons está na L_1 -ésima linha e C_1 -ésima coluna do tabuleiro.

A última linha contém dois inteiros L_2 e C_2 , indicando que a cidade B da competição está na L_2 -ésima linha e C_2 -ésima coluna do tabuleiro.

Saída

Seu programa deverá produzir uma única linha contendo um único inteiro W , a quantidade máxima de bombons que Christina pode levar de modo a chegar à cidade da competição em no máximo D dias.

Restrições

- $2 \leq L \leq 100$
- $2 \leq C \leq 100$
- $1 \leq D \leq 1\,000\,000\,000$
- $1 \leq p_{ij} \leq 10\,000$ ou $p_{ij} = -1$ para todos $1 \leq i \leq L$ e $1 \leq j \leq C$
- $1 \leq L_1 \leq L$ e $1 \leq C_1 \leq C$
- $1 \leq L_2 \leq L$ e $1 \leq C_2 \leq C$
- $(L_1, C_1) \neq (L_2, C_2)$
- É garantido que existe caminho entre A e B que não atravessa cidades com monstros

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (22 pontos):**
 - $L = C = 2$
 - $D \leq 100$
 - $p_{ij} \leq 100$ e $p_{ij} \neq -1$ para todos $1 \leq i \leq L$ e $1 \leq j \leq C$
 - $(L_1, C_1) = (1, 1)$
 - $(L_2, C_2) = (2, 2)$
- **Subtarefa 3 (27 pontos):** Todas as cidades sem monstros possuem a mesma dificuldade, ou seja, $p_{ij} = -1$ ou $p_{ij} = k$ (para alguma constante k) para todos $1 \leq i \leq L$ e $1 \leq j \leq C$.
- **Subtarefa 4 (20 pontos):** Existe um único caminho entre A e B que não atravessa cidades com monstros.
- **Subtarefa 5 (31 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 2 100 10 100 1 10 1 1 2 2	9

Exemplo de entrada 2	Exemplo de saída 2
3 4 37 2 2 2 2 2 -1 -1 2 2 2 2 2 3 3 1 1	4

Exemplo de entrada 3	Exemplo de saída 3
3 4 1000 1 2 3 4 -1 -1 -1 5 9 8 7 6 3 1 1 1	22

Estradas em Nlogônia

Nome do arquivo: `estradas.c`, `estradas.cpp`, `estradas.java`, `estradas.js` ou `estradas.py`

A Nlogônia é um reino muito distante onde vivem a rainha e suas N filhas. Cada filha da rainha governa uma das N cidades de Nlogônia. Atualmente, todas as cidades estão isoladas umas das outras, e não há nenhuma estrada entre qualquer par de cidades.

A rainha decidiu construir estradas para conectar as cidades do reino. Para isso, ela contratou a engenheira Margaret. Margaret pode escolher duas cidades A e B e construir uma estrada ligando essas duas cidades, desde que não exista ainda um caminho que conecte essas duas cidades usando as estradas já construídas. Estradas diferentes podem ter tamanhos diferentes.

Infelizmente, as filhas da rainha não se gostam, então elas não querem muito contato com suas irmãs. Assim, quando Margaret constrói uma estrada que conecta as cidades A e B , uma das duas filhas (a da cidade A ou a da cidade B) pede a Margaret para que nunca mais crie estradas ligando a sua cidade a uma outra cidade. Margaret sempre respeita o pedido, ou seja, **uma das duas cidades nunca terá outras estradas construídas no futuro**.

Neste reino, é chamado de *componente* um conjunto de cidades tal que há um caminho de estradas entre quaisquer duas cidades no conjunto e, além disso, não há caminho entre uma cidade no conjunto e outra que não está no conjunto. Note que uma componente pode mudar conforme novas estradas são construídas. A *distância* entre um par de cidades na mesma componente é a soma dos tamanhos das estradas no caminho entre elas.

Enquanto Margaret constrói as estradas, a rainha lhe fará várias perguntas para saber como está o andamento do trabalho. Em cada pergunta, a rainha escolhe uma cidade C e quer saber como está a situação da componente que possui a cidade C . Margaret deve responder com o valor da soma das distâncias entre todos os pares de cidades da componente que possui a cidade C .

Margaret pediu sua ajuda para responder as perguntas da rainha. Ela te entregou uma lista com Q linhas, onde cada linha contém a descrição de uma estrada que ela construiu ou a descrição de uma pergunta da rainha. Ajude Margaret a responder todas as perguntas corretamente!

Entrada

A primeira linha da entrada contém dois inteiros N e Q , o número de cidades na Nlogônia e o número de linhas na lista que Margaret entregou, respectivamente.

As próximas Q linhas contêm a lista de Margaret em ordem cronológica. Na i -ésima destas linhas, o primeiro inteiro, T_i , indica se a linha contém a descrição de uma estrada ou de uma pergunta:

- Se $T_i = 1$, a linha contém outros três inteiros, A_i , B_i e D_i , indicando que Margaret construiu uma estrada de tamanho D_i entre as cidades A_i e B_i .
- Se $T_i = 2$, a linha contém apenas um outro inteiro, C_i , indicando que a rainha deseja consultar a situação da componente que contém a cidade C_i .

Saída

Para cada pergunta da rainha, na ordem da entrada, seu programa deverá produzir uma linha com um único inteiro, a soma das distâncias entre todos os pares de cidades na componente da cidade escolhida pela rainha.

Restrições

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 200\,000$
- $T_i = 1$ ou $T_i = 2$ para todo $1 \leq i \leq Q$
- Se $T_i = 1$, $1 \leq A_i, B_i \leq N$
- Se $T_i = 1$, $1 \leq D_i \leq 1\,000\,000$
- Se $T_i = 2$, $1 \leq C_i \leq N$
- É garantido que toda construção de estrada satisfaz as condições do enunciado

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (26 pontos):**
 - $1 \leq N \leq 200$
 - $1 \leq Q \leq 200$
 - Para toda estrada construída, $D_i = 1$
- **Subtarefa 3 (22 pontos):**
 - $1 \leq N \leq 3\,000$
 - $1 \leq Q \leq 3\,000$
- **Subtarefa 4 (18 pontos):** As perguntas só serão feitas após todas as estradas serem construídas.
- **Subtarefa 5 (34 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
3 8	1
1 1 2 1	1
2 1	0
2 2	4
2 3	4
1 1 3 1	4
2 1	
2 2	
2 3	

Exemplo de entrada 2	Exemplo de saída 2
5 5 1 1 4 6 1 5 4 3 1 3 2 4 2 1 2 2	18 4

Exemplo de entrada 3	Exemplo de saída 3
7 12 1 2 5 3 2 5 1 3 5 4 2 3 1 1 4 3 2 1 1 1 7 6 2 1 1 5 6 2 2 3 1 6 7 2 2 4	3 14 3 18 27 168