Частина 1

Створюємо структуру Interval.

Структура має в собі три поля. a - ліва межа інтервалу, b - права межа інтервалу, t - тип інтервалу. Тип інтервалу визначається за допомогою переліку Typeinterval.

Структура має три конструктори. Конструктор за замовчуванням(Інтервал робиться від мінус нескінченності до плюс нескінченності), конструктор з параметрами та конструктор копіювання. Також структура має деструктор.

В струкрутрі ми перевантажуємо оператори вводи та виводу в потік, для оператору виводу ми також робимо вигляд інтервалу більш зрозумілим. Також ці перевантаженні оператори є дружніми функціями.

Функція перетину двох інтервалів працює за наступним алгоритмом:

Результат перетину двох інтервалів це максимальна з лівих координат та мінімальна з правих координат. Якщо цей результат є коректним, тобто ліва межа інтервалу менша за праву ми повертаємо його, а якщо ні то порожню множину(інтервал (0, 0)).

Функція об’єднання двох інтервалів працює за наступним алгоритмом:

Результат об’єднання двох інтервалів можливий тоді коли ліва межа другого інтервалу попадає в перший інтервал, тоді результатом об’єднання буде інтервал такого вигляду

(ліва межа першого інтервалу, максимум з правої межі першого та правої межі другого інтервалів). Якщо ж ця умова не виконалась ми просто повертаємо перший інтервал.

Функція різниці двох інтервалів це інтервал результату віднімання двох інтервалів.

Частина 2

Створюємо структуру SetIntervals.

Структура має в собі одне поля, вектор в якому будуть зберігатися інтервали.

Структура має три конструктори. Конструктор за замовчуванням(Вектор інтервалів порожній), конструктор в який ми передаємо вектор інтервалів та конструктор в який ми передаємо вектор пар дійсних чисел, а вже з них по черзі робимо інтервали та додаємо їх до нашого вектору інтервалів.

В струкрутрі ми перевантажуємо оператори вводи та виводу в потік, для оператору виводу ми також робимо вигляд інтервалу більш зрозумілим. Також ці перевантаженні оператори є дружніми функціями.

Функція перетину двох векторів інтервалів працює за наступним алгоритмом:

Ми беремо попарно інтервали з двох векторів інтервалів, знаходимо результат перетину цих двох інтервалів та цей результат додаємо до нового вектору інтервалів. Після того як ми візьмемо всі пари та виконаємо для них ці дії ми отримаємо вектор перетину двох векторів інтервалів.

Функція об’єднання двох векторів інтервалів працює за наступним алгоритмом:

Ми беремо перший інтервал з першого вектору інтервалів. Далі підтримуємо індекси інтервалів в векторах інтервалів. Поки в нас ці індекси не дійшли до кінців векторів інтервалів, ми намагаємося об’єднати інтервал, що ми взяли раніше спочатку з інтервалом з першого вектору, потім з другого, якщо це виходить ми їх об’єднуємо та оновлюємо індекс та інтервал. Якщо ні, то цей інтервал додається до вектору об'єднання, а ми беремо знову перший з якогось з векторів.

Функція різниці двох векторі інтервалів працює за наступним алгоритмом:

Ми беремо попарно інтервали з двох векторів інтервалів, знаходимо результат різниці цих двох інтервалів та цей результат додаємо до нового вектору інтервалів. Після того як ми візьмемо всі пари та виконаємо для них ці дії ми отримаємо вектор різниці двох векторів інтервалів.

Функція довжини вектору інтервалів працює так:

Ми знаходимо суму довжин всіх інтервалів з цього вектору та повертаємо це значення.

Частина 3

Створюємо структуру SquareInrquality

Структура має в собі чотири поля. a, b, c - коефіцієнт квадратної нерівності. t - тип квадратної нерівності. Тип квадратної нерівності визначається за допомогою переліку Type.

Структура має два конструктори. Конструктор за замовчуванням(Квадратна нерівність з коефіцієнтами 0, 0, 0 та типом дорівнює нулю) та конструктор з параметрами.

Функція getValue повертає значення квадратної нерівності в точці х.

Функція solve працює за наступним алгоритмом:

Спочатку ми знаходимо значення дискримінанту відповідного квадратного рівняння. Якщо дискримінант менше нуля, то наше рівняння не перитинає вісь Х, в отже немає розв’язків.

Якщо дискримінант дорівнює нулю, то в нас лише один розв'язок цього рівняння в точці х. Тоді ми відштовхуємось від нашого типу нерівності:

1. Якщо нерівність == 0. Розв'язок нерівності лише в точці х
2. Якщо нерівність != 0. Розв'язок нерівності - це інтервал від мінус нескінченності до х, та від х до плюс нескінченності.
3. Якщо нерівність > 0. Ми перевіряємо значення лівіше та правіше х, якщо якесь значення більше нуля то додаємо цей інтервал до розв'язку
4. Якщо нерівність >= 0. Ми перевіряємо значення лівіше та правіше х, якщо якесь значення більше нуля то додаємо цей інтервал до розв'язку, також додаємо ще й сам х.
5. Якщо нерівність < 0. Ми перевіряємо значення лівіше та правіше х, якщо якесь значення менше нуля то додаємо цей інтервал до розв'язку
6. Якщо нерівність <= 0. Ми перевіряємо значення лівіше та правіше х, якщо якесь значення менше нуля то додаємо цей інтервал до розв'язку, також додаємо ще й сам х.

Якщо дискримінант не дорівнює нулю, то в нас два розв'язки цього рівняння в точках х1 та х2. Вважаємо, що х1 меньше х2. Тоді ми відштовхуємось від нашого типу квадратної нерівності:

1. Якщо нерівність == 0. Розв'язок нерівності лише в точках х1, х2
2. Якщо нерівність != 0. Розв'язок нерівності - це інтервал від мінус нескінченності до х1, від х1 до х2 та від х2 до плюс нескінченності.
3. Якщо нерівність > 0. Ми перевіряємо значення лівіше х1, правіше х1 та лівіше х2, якщо якесь значення більше нуля то додаємо цей інтервал до розв'язку
4. Якщо нерівність >= 0. Ми перевіряємо значення лівіше х1, правіше х1 та лівіше х2, якщо якесь значення більше нуля то додаємо цей інтервал до розв'язку
5. Якщо нерівність < 0. Ми перевіряємо значення лівіше х1, правіше х1 та лівіше х2, якщо якесь значення менше нуля то додаємо цей інтервал до розв'язку
6. Якщо нерівність <= 0. Ми перевіряємо значення лівіше х1, правіше х1 та лівіше х2, якщо якесь значення менше нуля то додаємо цей інтервал до розв'язку