

БГУИР

Кафедра ИИТ

Отчет по лабораторной работе №2
Тема: «Алгоритмы операций над графиками и их реализация»

Выполнил:
Студент группы 421701 Федосов Егор.

Проверил:
Рудьман И.Д.

Минск
2024

Цель работы:

Научиться составлять алгоритм выполнения операций над графиками.

Задача:

Выполнить все основные операции над графиками.

Уточнение постановки задачи:

Ограничения на элементы множеств и графиков: элементы должны быть целыми числами в диапазоне $[-2^{31}; 2^{31}-1]$. Ограничений на область определения и область значений нет.

Определения и понятия, использовавшиеся в ходе выполнения работы:

1. График — это множество пар, т.е. множество, каждый элемент которого является парой или кортежем длины 2. Множество P называется графиком, если каждый его элемент — пара.

2. Областью определения графика P называется множество pr_1P (проекция на первую ось (ось абсцисс) данного графика).

3. Областью значения графика называется множество проекций на вторую ось (ось ординат) (pr_2P).

4. Инверсией графика P называют множество инверсий пар из P .

5. График P называется симметричным, если он наряду с любой своей парой содержит ее инверсию.

6. Пусть M — произвольное множество. Тогда считают ΔM — множество всех пар вида $\langle x, x \rangle$, где x присутствует во всем множестве M . ΔM является симметричным графиком и называется *диагональю*.

7. Композиция графиков P и Q — график $P \circ Q$, в котором упорядоченные пары удовлетворяют следующему свойству:

существует элемент z такой, что выполняется $\langle x, z \rangle \in P$ и $\langle z, y \rangle \in Q$: $P \circ Q = \{\langle x, y \rangle \mid \langle x, z \rangle \in P \text{ и } \langle z, y \rangle \in Q\}$.

Алгоритмы для упрощения реализации программы:

1. Проверка наличия некоторой пары $\langle a, b \rangle$ в графике P .
 - 1.1. Берем пару $\langle c, d \rangle$ графика P . Далее см. п. 1.2.
 - 1.1.1. Берем следующую пару $\langle c, d \rangle$ графика P .
 - 1.2. Проверяем пару $\langle c, d \rangle$ на предмет совпадения с парой $\langle a, b \rangle$
 - 1.2.1. Если пара $\langle c, d \rangle$ совпадает с парой $\langle a, b \rangle$ ($c=a$ и $d=b$), то см. п. 1.3.
 - 1.2.2. Если пара $\langle c, d \rangle$ не совпадает с парой $\langle a, b \rangle$ ($c \neq a$ и $d \neq b$) и не является последней в графике, см. п. 1.1.1.
 - 1.2.2.1. Если пара $\langle c, d \rangle$ не совпадает с парой $\langle a, b \rangle$ ($c \neq a$ и $d \neq b$) и является последней в графике P , то пара $\langle a, b \rangle$ не найдена в графике P .
 - 1.3. Пара $\langle a, b \rangle$ найдена в графике P .
2. Проверка наличия некоторого элемента E в множестве X .
 - 6.1. Берем элемент множества X . Далее смотреть пункт 6.2.
 - 6.1.1. Берем следующий элемент множества X .
 - 6.2. Проверяем данный элемент на совпадение с E .
 - 6.2.1. Если данный элемент совпадает с E , то переходим к пункту 6.3.
 - 6.2.2. Если он не совпадает с E и не является последним, то см. п. 6.1.1.
 - 6.2.3. Если элемент является последним, то элемент не найден.
 - 6.3. Элемент найден

Алгоритм реализации операций:

1. Операция инверсии графика P . Алгоритм реализации данной операции следующий:
 - 1.1. Создаем пустой график Q , в который в будущем будет помещен результат выполнения операции инверсии графика P .

- 1.2. Берем пару $\langle a, b \rangle$ графика P . Далее см. п. 1.3.
 - 1.2.1. Берем следующую пару $\langle a, b \rangle$ графика P .
- 1.3. Добавляем пару $\langle b, a \rangle$ к графику Q .
 - 1.3.1. Если пара последняя в графике P , см. п. 1.4.
 - 1.3.2. Иначе, см. п. 1.2.1.
- 1.4. Инверсия графика P составлена.

2. Операция нахождения композиции графиков P и Q .
Алгоритм следующий:

- 2.1. Создаем пустой график R , в котором в будущем будет помещен результат выполнения операции композиции.
- 2.2. Берем пару $\langle a, b \rangle$ графика P . Далее см. п. 2.2.2.
 - 2.2.1. Берем следующую пару $\langle a, b \rangle$ графика P .
 - 2.2.2. Берем пару $\langle c, d \rangle$ графика Q . Далее см. п. 2.2.2.2.
 - 2.2.2.1. Берем следующую пару $\langle c, d \rangle$ графика Q .
 - 2.2.2.2. Если $b \neq c$, то см. п. 2.2.2.1.
 - 2.2.2.2.1. Иначе проверяем наличие пары $\langle a, d \rangle$ в графике R (подробнее см. п. 1 раздела «Алгоритмы для упрощения реализации программы»)
 - 2.2.2.3. Если пара $\langle a, d \rangle$ не была найдена в графике R , то добавляем ее в график.
 - 2.2.2.4. Если пара $\langle c, d \rangle$ является последней парой в графике Q , то см. п. 2.2.3.
 - 2.2.2.4.1. Иначе, см. п. 2.2.2.1.
 - 2.2.3. Если пара $\langle a, b \rangle$ является последней парой в графике P , то см. п. 2.3.
 - 2.2.3.1. Иначе, см. п. 2.2.1.
- 2.3. Композиция графиков P и Q найдена.

3. Операция нахождения области определения графика G .
Алгоритм:

- 3.1. Создаем пустое множество D , в котором будет храниться область определения графика G .
- 3.2. Берем пару $\langle x, y \rangle$ из графика G .
 - 3.2.1. Берем следующую пару $\langle x, y \rangle$ из графика G .
 - 3.2.2. Проверяем наличие элемента x в множестве D (см. п. 2 раздела «Алгоритмы для упрощения реализации программы»).

3.2.2.1. Если элемент x не был найден, то добавляем его.

3.2.3. Если пара $\langle x, y \rangle$ не является последней, то см. п. 3.2.1.

3.3. Область определения графика G найдена.

4. Операция нахождения области значений графика G .
Алгоритм:

4.1. Создаем пустое множество E , в котором будет храниться область значений графика G .

4.2. Берем пару $\langle x, y \rangle$ из графика E .

4.2.1. Берем следующую пару $\langle x, y \rangle$ из графика E .

4.2.2. Проверяем наличие элемента y в множестве E (см. п. 2 раздела «Алгоритмы для упрощения реализации программы»).

4.2.2.1. Если элемент y не был найден, то добавляем его.

4.2.3. Если пара $\langle x, y \rangle$ не является последней, то см. п. 3.2.1.

4.3. Область значений графика G найдена.

5. Операция нахождения декартового произведения двух множеств A и B

5.1. Инициализация.

5.1.1. Создаём пустое множество C , в котором будем хранить элементы декартового произведения $A \times B$.

5.2. Заполнение первого множества A .

5.2.1. Пользователь выбирает способ заполнения множества A :

5.2.1.1. **Вручную:** пользователь вводит каждый элемент множества.

5.2.1.2. **Квадратами натуральных чисел в диапазоне:** пользователь задаёт диапазон $[A, B]$, в котором будут вычислены квадраты натуральных чисел.

5.3. Заполнение второго множества B .

5.3.1. Пользователь выбирает способ заполнения множества B аналогично множеству A :

5.3.1.1. **Вручную:** пользователь вводит каждый элемент множества.

5.3.1.2. **Квадратами натуральных чисел в диапазоне:** пользователь задаёт диапазон $[A, B]$, в котором будут вычислены квадраты натуральных чисел.

5.4. Вычисление декартового произведения.

5.4.1. Для каждого элемента a из множества A :

5.4.1.1. Для каждого элемента b из множества B :

5.4.1.1.1. Создаём пару (a,b) .

5.4.1.1.2. Добавляем пару (a,b) в множество C .

5.5. Вывод результатов.

5.5.1. Показываем пользователю составленные множества A и B .

5.5.2. Выводим все элементы декартового произведения C на экран.

Алгоритм выполнения программы:

1. Сначала пользователю предоставляется право выбрать операцию.

2. Далее, в зависимости от операции, пользователю будет необходимо ввести данные для выполнения выбранной им операции.

3. Затем происходит обработка введенного пользователем требования.

4. Результат выполнения выбранной пользователем операции выводится в консоль.

Вывод:

В ходе выполнения данной лабораторной работы я научился составлять алгоритм выполнения некоторых операций над графиками и сделал это, используя язык программирования C++.