Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

**Лабараторная работа №2**

На тему:

«Алгоритмы операцй над графиками и их реализация»

**Выполнил:**

Сечейко Н.В.

**Проверил:**

Рудьман И.О.

Минск 2024

*Цель*:

Научиться составлять алгоритм выполнения операций над графиками. Проверить выполнение составленного алгоритма в выбранной среде разработки на выбранном языке программирования(C++).

*Определения*:

**График** - это множество пар, то есть множество, каждый элемент которого является парой или кортежём длины 2. Множество называется графиком, если каждый элемент его есть пара.

**Кортеж** - упорядоченная последовательность/упорядоченное множество.

**Мощность множества** - характеристика множества, обобщающая понятие количества элементов конечного множества.

**Множество** - набор(совокупность) каких-либо неповторяющихся элементов.

**Область определения** графика – множество проекции на 1 ось данного графика.

**Множество значений** графика – множество проекции на 2 ось данного графика.

*Язык реализации действий над графиками*: C++

*Среда разработки программы*: visual studio

*Алгоритм выполнения:*

1. Запрашиваем от пользователя ввод количества пар в будущем графике(1) *a*.
2. Выполняем проверку на целочисленный ввод.
   1. Если результат округления введённого пользователем числа *a* не равен самом числу *a*, говорим пользователю, что ввод должен быть целочисленным и завершаем выполнение программ.
   2. Если результат округления введённого пользователем числа *a* равен самому числу *a,* переходим на следующий пункт.
3. Выполняем проверку на положительный ввод.
   1. Если введённое пользователем число *a* небольше нуля, сообщаем это пользователю и завершаем работу программы.
   2. Если введённое пользователем число *a* больше нуля, переходим на следующий пункт.
4. Задаём новое множество мощностью *a.*
5. Заполняем множество парами чисел(кортежами по два элемента).

\*На этом этапе множество, элементами которого являются пары чисел, становится графиком 1.

1. Запрашиваем от пользователя ввод количества пар в будущем графике(2) *b*.
2. Выполняем проверку на целочисленный ввод.
   1. Если результат округления введённого пользователем числа *b* не равен самом числу *b*, говорим пользователю, что ввод должен быть целочисленным и завершаем выполнение программ.
   2. Если результат округления введённого пользователем числа *a* равен самому числу *b,* переходим на следующий пункт.
3. Выполняем проверку на положительный ввод.
   1. Если введённое пользователем число *b* небольше нуля, сообщаем это пользователю и завершаем работу программы.
   2. Если введённое пользователем число *b* больше нуля, переходим на следующий пункт.
4. Задаём новое множество 2 мощностью *b.*
5. Заполняем множество парами чисел(кортежами по два элемента).

\*На этом этапе множество, элементами которого являются пары чисел, становится графиком 2.

1. Запрашиваем от пользователя выбор действия над графиком.
   1. Если пользователь выбирает «1. Инверсия» - переходим на пункт 12.
   2. Если пользователь выбиарет «2. Композиция» - переходим на пункт 13.
   3. Если пользователь выбирает «3. Проекция на оси» - переходим на пункт 14.
   4. Если пользователь выбирает «4. Декартово произведение» - перехоидм на пункт 15.
2. Инверсия графиков
   1. Выбираем первую(-ый) пару(кортеж) графика.
   2. Выбираем первый и второй элементы выбранного кортежа.
   3. Меняем местами выбранные элементы
   4. Если больше пар в графике нету, переходим на пункт 12.7.
   5. Выбираем следующую(-ий) пару(кортеж).
   6. Переходим на пункт 12.2.
   7. Изменённое множество будетя являться инверсией начального графика.
3. Композиция графиков.

Ввод: дано пустое множество 3, графики 1 и 2.

* 1. Выбрираем первую(-ый) пару(котреж) графика 1.
  2. Выбрираем первую(-ый) пару(котреж) графика 2.
  3. Если второй элемент выбранной пары графика 1 равен первому элементу выбранной пары графика 2, переходим на пунтк 13.10.
  4. Если пар в графике 2 больше нету, переходим к пункту 13.7.
  5. Берём следующую пару графика 2.
  6. Переходим на пункт 13.3.
  7. Если пар в графике 1 больше нету, переходим к пункту 13.17.
  8. Берём следующую пару графика 1.
  9. Переходим на пункт 13.2.
  10. Формируем кортеж, состоящий из двух элементов. Первый элемент – первый элемент выбранного кортежа 1 графика, второй элемент – второй элемент выбранного кортежа 2 графика.
  11. Если во множестве 3 есть элементы, выбираем первый из них. Если нет – переходим к пункту 13.15.
  12. Сравниваем сформированный кортеж с выбранной парой множества 3.
      1. Если кортежи равны – переходим к пункту 13.4.
      2. Если кортежи не равны – переходим к пункту 13.13.
  13. Если элементов во множестве 3 больше нету – переходим к пункту 13.15.
  14. Выбираем следующий элемент множества 3 и переходим к пункту 13.12.
  15. Добавляем сформированный кортеж во множество 3.
  16. Переходим на пункт 13.4.
  17. Полученный график 3 есть график композиции графиков 1 и 2.

**14.** Декартово произведение множеств.

Ввод: даны множества 1 и 2, дано пустое множество 3.

* 1. Выбираем первый элемент множества 1.
  2. Выбираем первый элемент множества 2.
  3. Формируем кортеж, состоящий из двух элементов. Первый элемент – выбранный элемент множества 1, второй – выбранный элемент множества 2.
  4. Если во множестве 3 нету элементов – переходим к пункту 14.9, если есть – выбираем первый элемент множества 3.
  5. Сравниваем сформированный кортеж с первой выбранной парой множества 3.
     1. Если кортежи совпадают – переходим к пункту 14.9.
     2. Если кортежи различаются – переходим к пункту 14.6.
  6. Если кортежей во множестве 3 больше нет – переходим к пункту 14.8.
  7. Выбираем следующую пару множества 3 и переходим к пункту 14.5.
  8. Добавляем сформированный кортеж во множество 3.
  9. Если во множестве 2 больше элементов нету – переходим к пункту 14.6.
  10. Выбираем следующий элемент множества 2 и переходим к пункту 14.3.
  11. Если во множестве 1 больше элементов нету – переходим к пункту 14.8.
  12. Выбираем следующий элемент множества 1 и переходим к пункту 14.2.
  13. Полученное множество 3 – результат декартового произведения.

**15.** Проекция на оси.

Ввод: дан график и пустое множество 3.

* 1. Выберем ось, на которую хотим сделать проекцию
     1. Если ось X(область определения) переходим к пункту 15.2.
     2. Если ось Y(множество значений) переходим к пункту 15.11.
  2. Выбираем первый кортеж графика.
  3. Выбираем первый элемент выбранного кортежа.
  4. Если во множестве 3 нету элементов – переходим к пункту 15.8, если есть – выбираем первый элемент множества 3.
  5. Сравниваем выбранный элемент выбранного кортежа с выбранным элементом множества 3.
     1. Если элементы равны – переходим к пунктку 15.9.
     2. Если элементы не равны – переходим к пункту 15.6.
  6. Если элементов во множестве 3 больше нет – переходим к пункту 15.8.
  7. Выбираем следуюзщй элемент множества 3 и переходим к пункту 15.5.
  8. Записываем выбранный элемент выбранного кортежа во множество 3.
  9. Если больше кортежей в графике нету - переходим к пункту 15.15.
  10. Выбираем следующий кортеж графика и переходим к пункту 15.3.
  11. Выбираем первый кортеж графика.
  12. Выбираем второй элемент выбранного кортежа.
  13. Если во множестве 3 нету элементов – переходим к пункту 15.17, если есть – выбираем первый элемент множества 3.
  14. Сравниваем выбранный элемент выбранного кортежа с выбранным элементом множества 3.
      1. Если элементы равны – переходим к пунктку 15.18.
      2. Если элементы не равны – переходим к пункту 15.15.
  15. Если элементов во множестве 3 больше нет – переходим к пункту 15.17.
  16. Выбираем следуюзщй элемент множества 3 и переходим к пункту 15.14.
  17. Записываем выбранный элемент выбранного кортежа во множество 3.
  18. Если больше кортежей в графике нету - переходим к пункту 15.20.
  19. Выбираем следующий кортеж графика и переходим к пункту 15.13.
  20. Полученное множество 3 – множество проекции графика на выбранную ось.

*Вывод:*

Мной были подробно спроектированны и описаны пункты алгоритма проведения различных действий над графиками.В результате работы я реализовал выполнение алгоритма на языке программирования C++.