МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД

**«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»**

**Кафедра програмного забезпечення комп'ютерних систем**

**Лабораторна робота №2**

з дисципліни: «ЯПЗ»

|  |  |
| --- | --- |
| Виконала: | ст. групи ПЗм-15-1м  Денисенко В.С. |
| Перевірив: | Козлов В.П. |

Дніпропетровськ

2016

**Тема:** Метрики сложности потока управления программ

**Цель:** По полученному в лабораторной работе № 1 алгоритму рассчитать метрики сложности потока управления программ (метрики Маккейба, Джилба, и граничных значений).

**Задание:** 8. Ввести массив А(100). Найти в нем первые десять элементов, которые больше 15. Вывести их значения и их номера в исходном массиве. Если их меньше десяти, вывести текст «Таких значений больше нет». Если их нет вообще, вывести текст «Таких значений нет».

**Исходный текст программы:**

//Коллекция для хранения резултатирующих значений (индекс => число)  
**LinkedHashSet<int[]>** pairs = **new** LinkedHashSet<>();

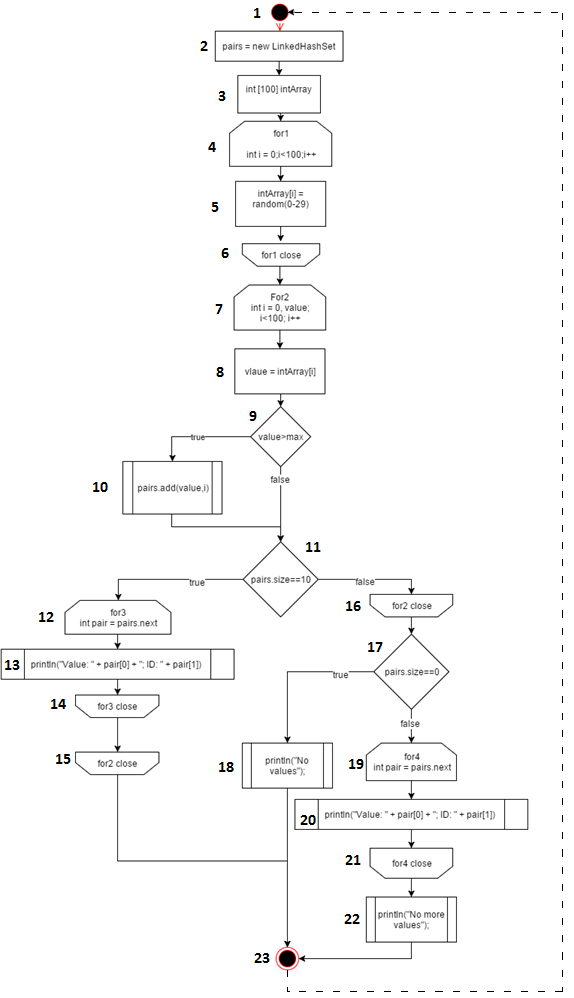
**int[]** intArray = **new int**[100]; //Массив исходных значений  
**int** max = 15; //Верхняя граница условия поиска  
  
//Наполнения исходного массива случайными значениями  
**for** (**int** i = 0; i < 100; i++) {  
 **System.*out*.print**( (intArray[i] = **new Random().**nextInt( 30 )) + "|" );  
}  
  
//Поиск значений соответствующих условию (max)  
**for** (int i = 0, value; i < 100; i++) {  
 value=intArray[i];  
 **if** ( value > max ) {

// Сохранение пары Индекс=>Число в коллекцию  
 pairs.add( **new int[]** {value, i} );

}

}  
   
 //Если количество найденных пар дошло необходимой отметки  
 **if** ( pairs.size() **==** 10 ) {   
 **System.*out*.println();**  
 //Вывод всех найденных пар (Индекс=>Число) в коллекции на экран  
 **for** (int[] pair **:** pairs) {  
 **System.*out*.println**( "Value: " + pair[0] + "; ID: " + pair[1] );   
 }  
 return;  
 }  
}  
**if** ( pairs.size() **==** 0 ) {  
 **System.*out*.println**( "\nТаких значений нет!" );  
} **else** {  
 **System.*out*.println();** **for** (int[] pair : pairs) {  
 **System.*out*.println**( "Value: " + pair[0] + "; ID: " + pair[1] );  
 }  
 **System.*out*.println**( "\nТаких значений больше нет!" );  
}

**Схема алгоритма:**



**Описание алгоритма:**

Задача программы выполнить поиск первых 10 значений массива, которые удовлетворяют условию – «значение > 15». Для этого выполнение программы можно разделить на 3 этапа:

1. Объявление и инициализация исходных данных, а именно переменные исходного, результирующего массивов, а также вспомогательные переменные.
2. Циклическая проверка исходных данных по заданному условию, для поиска значений, которые превышают поставленный задачей максимум (15), параллельно проверяя количество найденных значений удовлетворяющих условию. Если количество значений достигло десяти, то выполняется вывод всех найденных значений и прерывание выполнение программы.
3. Если же количество найденных значений не достигло необходимого количества, то выводятся все найденные значения и сообщение «Таких значений больше нет!». Если не было найдено ни одно удовлетворяющего значения, то выводится сообщение - «Таких значений нет».

**Расчет метрик потокоуправления:**

1. **Метрика Маккейба**

С помощью метрики Маккейба была проведена оценка трудоемкости тестирования программы. Метрика рассчитывается по формуле

где  **–** количество дуг графа, которое равняется **25,**

**–** количество вершин графа, которое равняется **23,**

**–** число компонентов связности графа, которое равняется **1.**

Подставив значение в формулу получаем:

Z(G) = 25 – 23 + 2\*1 = 4

Полученное значение показывает, что в работе алгоритма можно выделить 4 базовых линейно независимых контура:

1. (1 - 9(true)) – 10 – 11(true) – (12-15) – 23;
2. (1-9(false)) – 11(false) – 16 – 17(true) – 18 – 23;
3. (1-9(false)) – 11(false) – 16 – 17(false) – (19-23);
4. (1-9(false)) – 11(true) – (12-15) – 23;
5. **Метрика Джилба**

Расчет по метрике Джилба, определяет логическую сложность алгоритма(cl) по насыщенности условными операторами.

Результат метрики – соотношение количества условных операторов(СL= 3) к общем числу операторов программы (N = 23). CLI = 1.

Коэффициент логической сложности по метрике Джилба равен 0,14.

1. **Коэффициент скорректированной сложности**

Для нахождения коэффициента скорректированной сложности вершин графа было образовано 3 подграфа, начинающиеся с вершин выбора. В таблице 1.1 приведены расчеты по нахождению скорректированных сложностей вершин графа.

Таблица 1.1 – Скорректированные сложности вершин графа

|  |  |  |  |
| --- | --- | --- | --- |
|  | Номер вершины выбора | | |
| **Свойства подграфов** | **9** | **11** | **17** |
| **Номера вершин перехода** | 10 | 12,16 | 18,19 |
| **Номера вершин подграфа** | 10 | 12 - 22 | 18 - 22 |
| **Нижня граница подграфа** | 11 | 23 | 23 |
| **Сокректированая сложность вершины выбора** | 2 | 12 | 6 |

Абсолютной граничной сложностью графа является сумма скорректированных сложностей вершин этого графа. В таблице 1.2 приведены расчеты по нахождению коэффициента абсолютной граничной сложности программы.

Таблица 1.2 – Абсолютная граничная сложность графа

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |  |
| Скорректированная сложность | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 12 | 1 | 1 | 1 | 1 | 1 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 40 |

Подставив в формулу относительной граничной сложности графа полученные данные, получаем:

В таблице 2 приведены результаты расчетов метрик сложности потокоуправления программы для исходного графа.

Таблица 2 – Полученные коэффициенты метрик сложности потокоуправления

|  |  |
| --- | --- |
| Метрики сложности потокоуправления | |
| Метрика Маккейба - **Z(G)** | 4 |
| Абсолютная сложность программы - **CL** по **метрике Джилба** | 3 |
| Относительная сложность программы - **cl**  по **метрике Джилба** | 0,14 |
| Максимальный уровень вложенности условного оператора - **CLI** по **метрике Джилба** | 1 |
| Метрика граничных значений, **Sa** - абсолютная граничная сложность | 40 |
| Метрика граничных значений, **So** - относительная граничная сложность | 0,45 |

Выводы: в результате выполнения лабораторной работы были получены данные из расчетов нескольких метрик, указывающие на сложность потокоуправления исходной программы.