Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Индивидуальное практическое задание № 1

по дисциплине «Метрология, стандартизация и сертификация  
в информационных технологиях»

Вариант № 18

Выполнил студент: Сотников А.А.

группа 321702

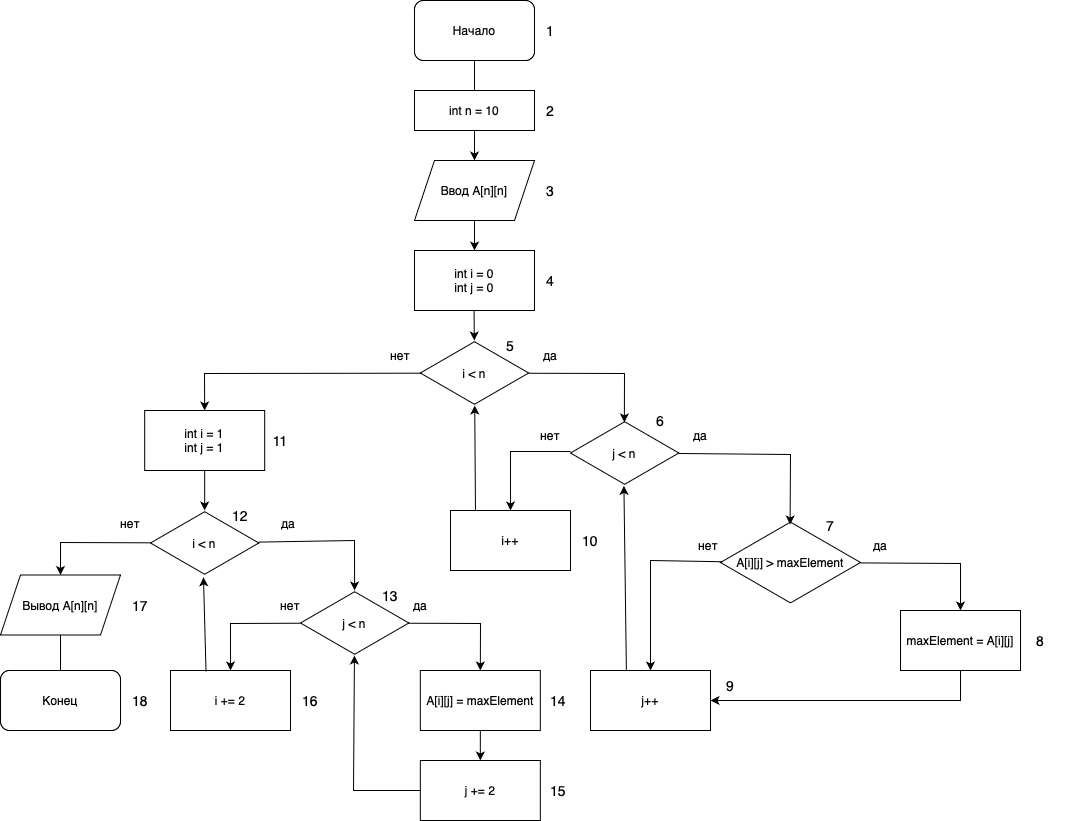
Зачетная книжка № 32170077

Минск 2024

**Цель:** разработать детализированную схему алгоритма, представленную в соответствии с положениями *ГОСТ 19.701–90*. По данному алгоритму рассчитать метрики сложности потока управления программ (метрики Маккейба, Джилба, максимальный уровень вложенности условного оператора и оператора цикла, метрику граничных значений).

**Условие:** Ввести массив *А(10, 10).* Заменить элементы массива, находящиеся на пересечении строк и столбцов с четными номерами, максимальным элементом данного массива. Вывести максимальный элемент, номера строки и столбца, в которых он находится в исходном массиве, исходный и результирующий массивы.

**Блок-схема:**



**Описание схемы алгоритма:**

1. **[Блок 1]** Начало: Запуск алгоритма.

2. **[Блок** **2]** Инициализация:

- Задается размерность массива n = 10.

3. **[Блок 3]** Ввод данных:

- Объявляется двумерный массив A[n][n].

- Вводятся значения элементов массива A[i][j].

4. **[Блок 4]** Инициализация переменных:

- i = 0 — для обхода строк массива.

- j = 0 — для обхода столбцов массива.

5. **[Блок 5]** Проверка условия i < n:

- Если i < n (есть строки для обработки), выполняется следующий блок.

- Если i ≥ n, переход к **[Блоку 10].**

6. **[Блок 6]** Сброс j на 0:

- Устанавливается j = 0 для начала обхода текущей строки.

7. **[Блок 7]** Проверка условия j < n:

- Если j < n (есть столбцы для обработки), выполняется следующий блок.

- Если j ≥ n, переход к **[Блоку 9].**

8. **[Блок 8]** Поиск максимального элемента:

- Если A[i][j] > maxElement, то maxElement = A[i][j].

- Увеличивается j на 1, переход к **[Блоку 7].**

9. **[Блок 9]** Переход к следующей строке:

- Увеличивается i на 1, переход к **[Блоку 5].**

10. **[Блок 10]** Модификация массива:

- Для всех чётных строк i, пока i < n, выполняется:

- Заменяются элементы строки A[i][j] на maxElement.

- Увеличивается i на 2.

11. **[Блок 11]** Вывод массива:

- Выводится обновленный массив A[n][n].

12. **[Блок 12]** Конец: Завершение работы алгоритма.

**Расчёт метрики Маккейба:**

Z(G) = e – υ + 2p

Где:

* e — число дуг ориентированного графа;
* υ — число вершин;
* p — количество компонент связности

Согласно нашему графу:

* e = 22;
* υ = 18;
* p = 1.

Z(G) = 22 − 18 + 2 ⋅ 1 = 5

**Базовые независимые пути:**

1. 1 – 2 – 3 – 4 – 5(да) – 6(да) – 7(да) – 8 – 9 – 6
2. 1 – 2 – 3 – 4 – 5(да) – 6(да) – 7(нет) – 9 – 6
3. 1 – 2 – 3 – 4 – 5(да) – 6(нет) – 10 – 5
4. 1 – 2 – 3 – 4 – 5(нет) – 11 – 12(да) – 13(да) – 14 – 15 – 13
5. 1 – 2 – 3 – 4 – 5(нет) – 11 – 12(да) – 13(нет) – 16 – 12
6. 1 – 2 – 3 – 4 – 5(нет) – 11 – 12(нет) – 17 – 18

**Расчёт метрики Джилба:**

Определяет логическую сложность программы как насыщенность программы условными операторами IF–THEN–ELSE и операторами цикла. Обычно используются два вида метрики Джилба:

*CL* – количество условных и циклических операторов, характеризующее абсолютную сложность программы;

*cl* – насыщенность программы условными и циклическими операторами, характеризующая относительную сложность программы;

*CLI* – максимальный уровень вложенности условного и циклического оператора;

Согласно нашим данным:

*CL = 5*

*cl = CL / (18 - 2) = 5 / 16 ≈ 0,3125*

*CLI = 2*

**Свойства подграфов программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Свойства подграфов программы | Номер вершины выбора | | | | |
| 5 | 6 | 7 | 12 | 13 |
| Номера вершин перехода | 6, 11 | 7, 10 | 8, 9 | 13, 17 | 14, 16 |
| Номера вершин подграфа | 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 | 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 | 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 | 13, 14, 15, 16, 17 | 14, 15, 16, 17 |
| Номер нижней границы подграфа | 13 | 9 | 9 | 17 | 16 |
| Скорректированная сложность вершины выбора | 12 | 11 | 10 | 5 | 4 |

**Скорректированные сложности вершин графа программы:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины графа  программы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | Sa |
| Скорректирован-ная сложность вершины графа | 1 | 1 | 1 | 1 | 12 | 11 | 10 | 12 | 11 | 12 | 6 | 5 | 4 | 5 | 4 | 6 | 1 | 1 | 104 |

***S0*** *= 1 – (18 - 1) / 104 =**0,154*;

**Метрики сложности потока управления программ:**

|  |  |
| --- | --- |
| Метрики сложности потока управления | Значение |
|
| Метрика Маккейба ***Z*(*G*)** | 5 |
| Абсолютная сложность программы ***CL***по метрике Джилба | 5 |
| Относительная сложность программы ***cl*** по метрике Джилба | 0,3125 |
| Максимальный уровень вложенности условного оператора ***CLI***по метрике Джилба | 2 |
| Метрика граничных значений ***Sa*** (абсолютная граничная сложность программы) | 104 |
| Метрика граничных значений ***Sо*** (относительная граничная сложность программы) | 0,154 |

**Вывод:**

Разработана схема алгоритма по ГОСТ 19.701–90 с учетом метрик сложности потока управления. Предусмотрен вывод всех входных и выходных данных, что позволяет улучшить анализ программы и её оптимизацию.