Реализация решения

Текст программы для реализации возможного решения поставленной задачи, разработанной с использованием языка программирования С#, приведен на рис. 2.30.

|  |  |
| --- | --- |
| Номера строк | Строки программы |
| 1 | namespace Ex |
| 2 | { |
| 3 | class Перестановка |
| 4 | { |
| 5 | static void Main() |
| 6 | { |
| 7 | int n, m, mp, iiomI, hom2, i,j; |
| 8 | double),] a; |
| 9 | double b; |
| 10 | ConsoleKeylnfo клавиша; |
| 11 | do |
| 12 | { |
| 13 | Console.ClearO; |
| 14 | Console.Write("CKoabKO строк:"); |
| 15 | n = int.Parse(Console.ReadLine()); |
| 16 | Console.Write("CKOBbKO столбцов: "); |
| 17 | m = int.Parse(Console.ReadLineO); |
| 18 | a = new double[n, m]; |
| 19 | for (i = 0; i < n; i++) |
| 20 | for (j = 0;.j < m; j++) |

|  |  |
| --- | --- |
| 21 | { |
| 22 | Console.Write("3neMeHT[{0},{l}]:", i,j); |
| 23 | a[i, j] = double.Parse(Console.ReadLine()); |
| 24 | } |
| 25 | Console.WriteLine("\nMcxoAHafl таблица"); |
| 26 | for (i = 0; i < n; i++, Console. WriteLineO) |
| 27 | for (j = 0; j < m; j++) |
| 28 | Console.Write(“ {0,6:f2}”, a[i, j]); |
| 29 | mp = n/2; ' |
| 30 | for(HOMl=0,HOM2=n-l; ном1<тр; ном1++,ном2—) |
| 31 | t  i  6  V  о  11  \*o  <2 |
| 32 | { |
| 33 | Ь = а[ном1,Л; |
| 34 | а[ном1, j] = а[ном2, j]; |
| 35 | а[ном2, j] = b; |
| 36 | } |
| 37 | Console. WriteLine("\nTa6nnL;a после перестановки строк"); |
| 38 | for (i = 0; i < n; i++, Console.WriteLineO) |
| 39 | for (j = 0; j < m; j++) |
| 40 | Console.Write((“ {0,6:f2}”, a[i, j]); |
| 41 | Console.\УгкеЬте("\пДля выхода нажмите клавишу ESC"); |
| 42 | клавиша = Console.ReadKey(true); |
| 43 | } while (клавиша.Кеу!= ConsoleKey.Escape); |
| 44 | } |
| 45 | } |
| 46 | 1 |

Рис. 2.30. Текст программы «Замена строк матрицы»

Алгоритм решения задачи выглядит так, как показано на рис. 2.31. Оценка алгоритмической сложности

Граф потока управления приведен на рис. 2.32. Тонированные вершины обозначают операторы ветвления.

Проведем оценку алгоритмической сложности графа по первому критерию. Определим минимальный набор маршрутов, проходящих через каждый оператор ветвления и по каждой дуге.

Для составленного графа можно выделить один маршрут:

ml: 1-2-3-4-5-4-6-3-7-8-9-10-9-11-8-12-13-14^15-14-16-

-13-17-18^9-20-19-21-18^22-2-3-7-8-12-13-17-18-22-23;

Р\ =22.

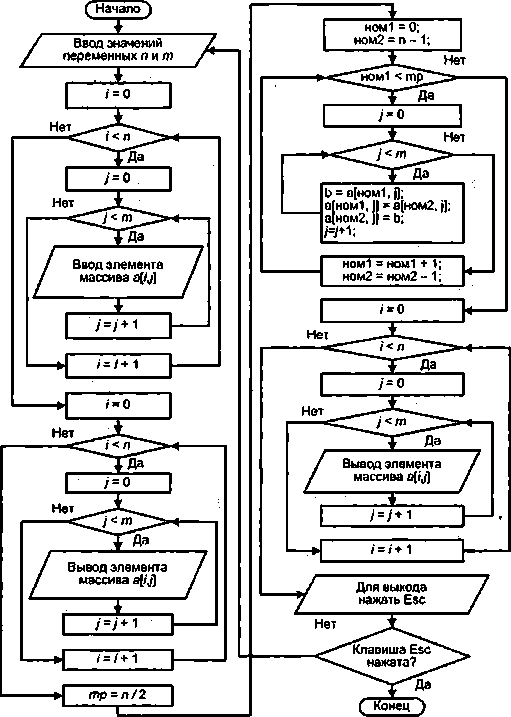


Рис. 2.31. Алгоритм замены строк матрицы

Выделенный маршрут проходит по всем вершинам ветвления и хотя бы один раз по каждой дуге графа управления потоком рассматриваемой задачи, соответствует требованиям первого критерия.

В перечне выделенного маршрута номера вершин ветвления выделены полужирным шрифтом с подчеркиванием. Количество маршрутов - 1, количество вершин ветвления в маршруте 22, тогда

S\ ~Р\ — 22.

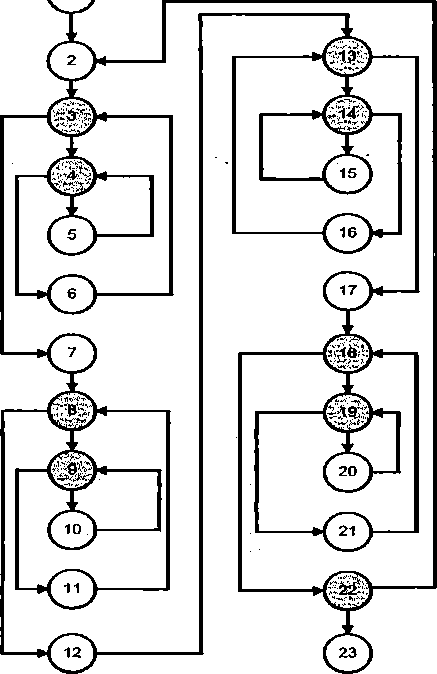


Рис. 2.32. Граф потока управления программой замены строк матрицы

Необходимо отметить, что уровень сложности данной программы достаточно высокий.

Проведем оценку алгоритмической сложности по второму критерию. Необходимо определить число проверок каждого линейно независимого цикла и линейно независимого ациклического участка программы. Количество проверок определяется цикпоматическим числом графа. Число вершин ветвления в составленном графе составляет 9, отсюда Z = па + 1=9+1 = 10.

Таким образом, общее число циклических и ациклических участков составляет 10. Выделим маршруты на полученном графе:

* ациклические маршруты:

т 1: 1-2-3-4-5-4-6-3-7-8-9-10-9-11-8-12-13-14-15-14-16- -13-17-18^9-20-19-21-18-22-23; рх = 17;

* циклические маршруты:

m2: 4—5; р2= 1;

m3:3 4 5 4 6; р2 — 3;

т4: 9—10;р4 = 1;

т5: 8-9-10-9-11;/?5= 3;

тб: 14-15: рв = 1;

ml: 13-14-15-14-16; р7= 3;

/и8:19-20: р\* = П

ш9:18-19-20-19-2\;р9 = 3;

/иЮ: 2-3-7-8-12-13-17-18-22;р10= 5.

Тестирование программы по указанным маршрутам позволит проверить все операторы ветвления программы. Метрика структурной сложности определяется по следующему соотношению:

s2 = Рх+ Р2 + РЗ + Р4 + Р5 + Рь + Р7 + Р» + Р9 + Рю =

= 17+1+3 + 1 + 3 + 1+ 3 + 1 + 3 + 5 = 38.

Для организации автоматического анализа заданного графа по второму критерию с помощью вычислительных средств построим матрицы смежности и достижимости.

Напомним, что матрица смежности представляет собой квадратную матрицу, размер которой определяется количеством вершин графа. Полученный граф имеет 23 вершины, следовательно, матрица смежности будет иметь размер 23x23. При заполнении матрицы следует придерживаться следующего правила: для дуги, соединяющей вершину 1 с вершиной 2 (см. рис. 2.32), в первый столбец и вторую строку матрицы смежности записываем 1 (табл. 2.5). Номер столбца соответствует номеру вершины, из которой выходит дуга, номер строки соответствует номеру вершины, в которую входит дуга. Таким образом заполняем матрицу для всех дуг управляющего графа.

Для выделения маршрутов можно использовать матрицу достижимости, которая представляет собой для полученного графа управления квадратную таблицу размером 23x23. Номер столбца матрицы определяет номер вершины графа, из которого возможно достижение других вершин этого же графа, используя цикличные и ацикличные маршруты.

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

Номера строк определяют номера достижимых вершин графа управления. Из вершины 1 возможно достичь вершины со 2-й по 23-ю, т. е. достижимы все вершины графа (см. рис. 2.32). Для первого столбца матрицы достижимости строки, начиная со второй, заполняются единицами (табл. 2.6). Далее заполняются столбцы для остальных вершин аналогично. Выделенные диагональные элементы матрицы определяют номера вершин, которые входят в состав циклических маршрутов. Идентичные строки матрицы определяют номера вершин, которые входят в состав ацикличных маршрутов.

Диагональные элементы, выделенные курсивом для различных вариантов шрифтов, соответствуют mlO: 2-3-7-8-12-13-17-18-22. Диагональные элементы, выделенные курсивом с подчеркиванием, соответствуют m3: Э^-4-5-4-6. На указанном участке элементы, выделенные черной заливкой, соответствуют m2:4-5. Выделенные элементы полужирным курсивом соответствуют т5\ 8-9-10-9-11.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | и | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | I | 1 | 1 | I | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |
| 3 | 1 | 1 | I | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 4 | 1 | 1 | 1 ВЯ 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 5 | 1 | 1 | 1 | 1 Ш 1 | | | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 7 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |
| 8 | 1 | 1 | 1 |  |  | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 | 1 | 1 | 1 | 1 |  |
| 9 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 ЕЯ 1 | | | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 | 1 | 1 | 1 |  |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 ИЯ 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 |  |
| 11 | 1 | 1 |  | 1 |  | : 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 |  | 1 | 1 |  |
| 12 | 1 | 1 |  | 1 |  | 1 | 1 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 |  | 1 |  |  |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 |  |  |
| 14 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 15 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 га 1 | | | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 16 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 17 | 1 |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 |  |  | 1 | 1 |  |
| 18 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | I | 1 | 1 | 1 |  |  | 1 |  |  | 1 |  |
| 19 | 1 | 1 | 1 |  |  |  | 1 | 1 | 1 | 1 |  | 1 | 1 | г |  | 1 | 1 | 1 ЕИ 1 | | | 1 | 1 |  |
| 20 | 1 |  | 1 |  | 1 | 1 | 1 |  |  | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 |  | 1 га 1 | | | 1 |  |
| 21 | 1 |  | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | I | 1 | 1 |  | 1 |  | 1 | 1 | 1 |  | 1 |  |
| 22 | 1 | 1 | 1 | I | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 |  | 1 | 1 |  |
| 23 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | '1 |  |

На указанном участке элементы, выделенные темной заливкой, соответствуют т4: 9-10. Диагональные элементы, выделенные полужирным курсивом с подчеркиванием, соответствуют ml: 13—14—15— 14-16. Залитые ячейки темным цветом на указанном участке соответствуют тб: 14-15. Диагональные элементы, выделенные курсивом иного шрифта, соответствуют т9: 18-19-20-19-21. Ячейки на указанном участке, залитые темным цветом, соответствуют т8: 19-20. Все строки матрицы достижимости идентичны и включают все номера вершин ациклического маршрута.

Проведем оценку алгоритмической сложности по третьему критерию. В соответствии с третьим критерием необходимо выделить все реально возможные маршруты управления:

ml: 1—2—3—4—5—4—6—3—7—8—9—10-9-11 -8-12-13-14-15-14-16- -13-17-18-49-20-19-21-1^-22-23; />,=1б; m2: 1-2-3-7-8-12-13-17-18-22-2-3-4-6-3-7-8-9-11-8-12-13- -14-16-13-17-18-19-21-18-22-23; р2=18;~ ““ “

m3: 1-2-3-4-5-4-6-3-7-8-9-10-9-11-8-12-13-14-15-14-16- -13-17-18-19-20-19-21-18-22-2-3-4-6-5-7-8-9— 11 12

—13—14—16—13—17—18—19—21—18—22—23: Pi=29. ~

Оценку структурной сложности программы проводим по следующему соотношению:

$з= Pt + Pi + Рз= 16 + 18 + 29 = 63.

Исходя из полученных результатов расчета метрик структурной сложности по первому (S\ = 22), второму (S2 = 38) и третьему (53 = 63) критериям выделения маршрутов можно сделать вывод, что программа, характеризуемая представленным графом управления, имеет высокую алгоритмическую сложность, так как количество используемых в тексте операторов условий 9, для проверки которых необходимо провести от 22 до 63 тестовых вариантов исходных данных.

Оценим алгоритмическую сложность программы на основе метрики Маккейба. В соответствии с теорией Маккейба сложность алгоритма оценивается величиной цикломатического числа, которая определяется по следующему соотношению: Z = т — п + 2, где т - количество дуг управляющего графа, построенного на основе алгоритма программы, п - количество вершин графа. В соответствии с управляющим графом от = 30 (см. рис. 2.32), п = 23, тогда цикломатическое число равно: .

Z = от - и + 2 = 30 - 23 + 2 = 9.

Таким образом, в соответствии со значением цикломатического числа (Z = 9) в полученном графе управления программой можно выделить девять независимых контуров, которые определяют девять управляющих маршрутов, ведущих из начальной вершины в конечную! Значение цикломатического числа для полученного графа не превышает значения 10, но очень близко к нему, что говорит о высокой сложности разработанного алгоритма, которая близка к критической.

1. Задача «Объединение аргументов командной строки»

Из принятых аргументов командной строки сформировать новую строку путем объединения всех аргументов командной строки. Каждый аргумент командной строки сортируется посимвольно по убыванию в машинном алфавите. В объединенной строке аргументы командной строки разделяются символом двоеточия:

* командная строка: exl.exe 1234 89 67;
* объединенная строка: 4321:98:76.

Запуск исполняемого файла должен осуществляться через главное меню операционной системы (кнопка «Пуск» —> «Выполнить»).

Разработать программу. В соответствии с разработанной программой составить блок-схему алгоритма решения задачи. На основании блок-схемы сформировать граф потока управления программой и оценить ее алгоритмическую сложность.

Реализация решения

Текст программы для реализации возможного решения поставленной задачи, разработанной с использованием языка программирования С#, приведен на рис. 2.33.

|  |  |
| --- | --- |
| Номера строк | Строки программы |
| 1 | using System; |
| 2 | namespace EX 1 |
| 3 | { |
| 4 | class Program |
| 5 | { |
| 6 | static void Main(string[] args) ' |
| 7 | { |
| 8 | string union; |
| 9 | char[] word; |
| 10 | int i; |
| 11 | for (i = 0; i < args.Length; i++) |
| 12 | { |
| 13 | word = new char[args[i].Length]; |
| 14 | args[i].CopyTo(0, word, 0, args[i].Length); |
| 15 | Array.Sort(word, 0, word.Length); |
| 16 | Array.Reverse(word, 0, word.Length); |
| 17 | args[i] = new string(word); |
| 18 | } |
| 19 | union = string.Join(":",args); |
| 20 | Console. WriteLine(union); |
| 21 | Console.ReadKeyO; |
| 22 | } |
| 23 | } |
| 24 | } |

Рис. 2.33. Текст программы объединения аргументов командной строки

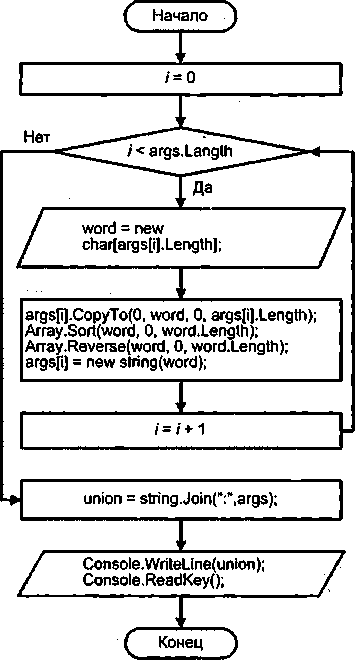


Рис. 2.34. Управляющий граф программы объединения аргументов

Алгоритм решения задачи приведен на рис. 2.34.

Оценка алгоритмической сложности

Граф потока управления приведен на рис. 2.35. Тонированная вершина обозначает оператор ветвления.

Проведем оценку алгоритмической сложности графа по первому критерию. Определим минимальный набор маршрутов, проходящих через каждый оператор ветвления и по каждой дуге.

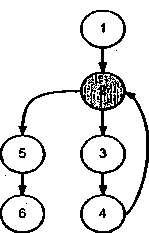


Рис. 2.35. Управляющий граф программы объединения аргументов

Для составленного графа можно выделить один маршрут: ml: 1-2-3-4-2-5-6; pi = 1.

Выделенный маршрут проходит по всем вершинам ветвления и хотя бы один раз по каждой дуге графа управления потоком рассматриваемой задачи, что соответствует требованиям первого критерия.

В перечне выделенного маршрута номер вершины ветвления выделен полужирным шрифтом с подчеркиванием. Количество маршрутов - 1, количество вершин ветвления в маршруте - 1. .

Уровень сложности алгоритма по первому критерию выбора маршрутов равен:

б) = р\ = 1.

Алгоритм данной программы весьма прост.

Проведем оценку алгоритмической сложности по второму критерию выбора маршрутов. Необходимо определить число проверок каждого линейно независимого цикла и линейно независимого ациклического участка программы. Количество проверок определяется цикло- матическим числом графа, которое определяется следующим соотношением: Z = пв + 1, где пв - число вершин ветвления. Число вершин ветвления в составленном графе составляет 1, отсюда

Z = «„+ 1 = 1 + 1 = 2.

Таким образом, общее число циклических и ациклических участков составляет 2. Выделим маршруты на полученном графе:

* ациклический маршрут: ml: 1-2-3^4-2-5-6;р\ = 1;
* циклический маршрут: m2: 2-3-4; р2 = 1.

Тестирование программы по указанным маршрутам позволит проверить все операторы ветвления программы. Метрика структурной сложности определяется по следующему соотношению:

S2= Р\+ Рг= 1 + 1 = 2.

Для организации автоматического анализа заданного графа по второму критерию с помощью вычислительных средств построим матрицы смежности и достижимости.

Напомним, что матрица смежности представляет собой квадратную матрицу, размер которой определяется количеством вершин графа. Полученный граф имеет 6 вершин, следовательно, матрица смежности будет иметь размер 6x6. При заполнении матрицы следует придерживаться следующего правила: для дуги, соединяющей вершину 1 с вершиной 2 (см. рис. 2.35), в первый столбец и вторую строку матрицы смежности записываем 1 (см. табл. 2.7). Номер столбца соответствует номеру вершины, из которой выходит дуга, номер строки соответствует номеру вершины, в которую входит дуга. Таким образом заполняем матрицу для всех дуг управляющего графа.

Таблица 2.7. Матрица смежности

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 |  |  |  |  |  |  |
| 2 | 1 |  |  | 1 |  |  |
| 3 |  | 1 |  |  |  |  |
| 4 |  |  | 1 |  |  |  |
| 5 |  | 1 |  |  |  |  |
| 6 |  |  |  |  | 1 |  |

Для выделения маршрутов можно использовать матрицу достижимости, которая представляет собой для полученного графа управления квадратную таблицу размером 6x6. Номер столбца матрицы определяет номер вершины графа, из которого возможно достичь другие вершины этого же графа, используя цикличные и ацикличные маршруты. Номера строк определяют номера достижимых вершин графа управления. Из вершины 1 возможно достичь вершины со 2-й по 6-ю, т. е. достижимы все вершины графа (см. рис. 2.35). Для первого столбца матрицы достижимости строки, начиная со второй, заполняются единицами (см. табл. 2.8). Далее заполняются столбцы для остальных вершин аналогичным образом. Выделенные диагональные элементы матрицы определяют номера вершин, которые входят в состав циклических маршрутов. Идентичные строки матрицы определяют номера вершин, которые входят в состав ацикличных маршрутов.

Таблица 2.8. Матрица достижимости

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 |  |  |  |  |  |  |
| 2 | 1 И1 | | 1 | 1 |  |  |
| 3 | 1 | 1 |  | 1 |  |  |
| 4 | 1 | '1 ■■■■■' | 1 |  |  |  |
| 5 |  | Г ■: | 1 | 1 |  |  |
| 6 | 1 | 1 | 1 | 1 | 1 |  |

Диагональные элементы, выделенные темным цветом, соответствуют циклическому маршруту m2: 2-3-4. В шестой строке определены номера столбцов, соответствующие номерам вершин графа, входящих в состав ациклического маршрута т\: 1 —2—3—4—2-5-6.

Сложность алгоритма по второму критерию отбора маршрутов невелика {S2= 2).

Проведем оценку алгоритмической сложности по третьему критерию, в соответствии с которым необходимо выделить все реально возможные маршруты управления:

т\: 1-2-3-4-2-5-6;рг=1; m2: 1—2-5-6;/?2=1.

Оценку структурной сложности программы рассчитываем по следующему соотношению:

5з =Р)+Р2= 1 + 1=2.

Исходя из полученных результатов расчета метрик структурной сложности по первому (Si = 1), второму (Л = 2) и третьему (S3 = 2) критериям выделения маршрутов можно сделать вывод, что программа, характеризуемая сформированным графом управления, имеет весьма низкую алгоритмическую сложность, так как используется всего один оператор условий, для проверки которых необходимо провести 2 тестовых варианта исходных данных.

Оценим алгоритмическую сложность программы на основе метрики Маккейба. В соответствии с теорией Маккейба сложность алгоритма оценивается величиной цикломатического числа, которая оп-

ределяется по следующему соотношению:^ = т - п + 2, где т - количество дуг управляющего графа, построенного на основе алгоритма программы, п - количество вершин графа. В соответствии с полученным управляющим графом т = 6, п = 6, тогда цикломатическое число равно: ,

Z = m-n + 2 = 6-6 + 2 = 2.

Таким образом, в соответствий со значением цикломатического числа (Z = 2) в полученном графе управления программой можно выделить два независимых контура, которые определяют два управляющих маршрута, ведущих из начальной вершины в конечную. Значение цикломатического числа для полученного графа весьма далеко от предельного значения 10, что говорит о низком уровне сложности разработанного алгоритма.

1. Задача «Проверка простого числа»

Программе с клавиатуры задается число. Необходимо разработать алгоритм для комплексного решения двух независимых задач:

* определить, является ли заданное число простым;
* определить количество цифр, составляющих заданное число.

По разработанному алгоритму построить граф потока управления

и оценить сложность алгоритма программы по управляющему графу.

Реализация решения

Алгоритм решения задачи может выглядеть так, как показано на рис. 2.36, а граф потока управления приведен на рис. 2.37.

Тонированные вершины обозначают операторы ветвления.

Оценка алгоритмической сложности

Проведем оценку алгоритмической сложности графа по первому критерию. Определим минимальный набор маршрутов, проходящих через каждый оператор ветвления и по каждой дуге.

Первый маршрут:

ml: 1 —2—3—4—5—4—5—6— 10—1Л—2—3—7—8—9—7—8—9— 10—34,—12; /7, = 8.

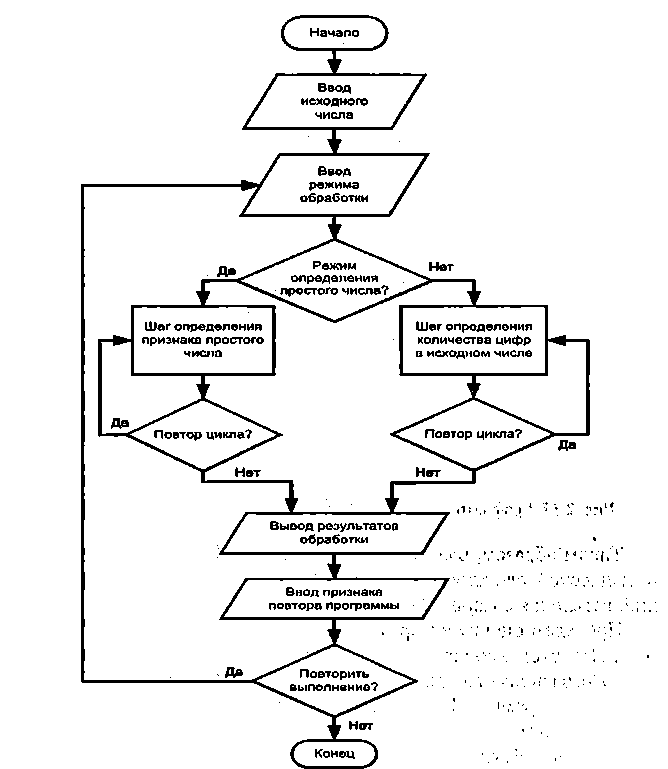


Рис. 2.36. Алгоритм решения задачи о простом числе

Данный маршрут проходит по всем вершинам ветвления и хотя бы один раз по каждой дуге графа, представленного на рис. 2.36), что соответствует требованиям первого критерия. В перечне участков маршрута номера вершин ветвления выделены полужирным шрифтом с Подчеркиванием/Количество таких вершин в маршруте составляет 8.

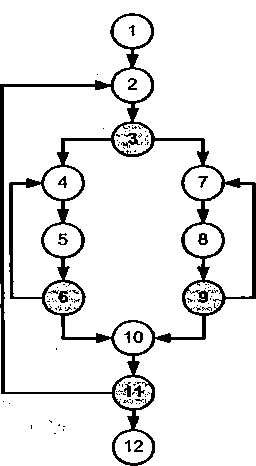


Рис. 2.37. Граф потока управления задачи определения простого числа

\

Таким образом, в соответствии с первым критерием оценки алгоритмической сложности количество маршрутов 1, количество вершин ветвления в маршруте 8, т. е. Si =pi = 8.

Проведем оценку алгоритмической сложности по второму критерию. При этом необходимо определить число проверок каждого линейно независимого цикла и линейно независимого ациклического участка программы. Количество проверок определяется цикломати- ческим числом графа, которое определяется следующим соотношением: Z = пв + 1, где пв - число вершин ветвления.

Число вершин ветвления в заданном графе составляет 4, отсюда

Z = +1 = 4+1 = 5.

Таким образом, общее число циклических и ациклических участков составляет 5. Выделим эти маршруты в заданном графе:

• ациклические маршруты:

ml: 1-2-3-4-5-6-10-14-12; р\ = 3; m2: 1-2-3-7-8-9-10-11-12; р2 = 3;

• циклические маршруты:

m3: 2-3-4—5-6-10-П;ръ = 3; ш4: 4—5-6; рл = 1; ш5: 7—8—9; ps = 1.

Тестирование программы по указанным маршрутам позволит проверить все операторы ветвления программы. Метрика структурной сложности определяется по следующему соотношению:

= Ру + Р2 + Р2 + р$ + р$ = 3 + 3 + 3 + 1 + 1 — 11.

Для организации автоматического анализа заданного графа по второму критерию с помощью вычислительных средств построим матрицы смежности и достижимости.

Напомним, что матрица смежности представляет собой квадратную матрицу, размер которой определяется количеством вершин графа. Заданный граф имеет 12 вершин, следовательно, матрица смежности будет иметь размер 12x12. При заполнении матрицы следует придерживаться следующего правила: для дуги, соединяющей вершину 1 с вершиной 2 (см. рис. 2.37) в первый столбец и вторую строку матрицы смежности записываем 1. Номер столбца соответствует номеру вершины, из которой выходит дуга, а номер строки - номеру •вершины, в которую входит дуга. Таким же образом заполняем матрицу для всех дуг управляющего графа (табл. 2.9).

Таблица 2.9. Матрица смежности для задачи простого числа

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 |  |  |  |  |  |  |  |  | ■ ■ 4 |  | •! ' |  |
| 2 | 1 |  |  |  |  |  |  |  |  |  | 1 |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  | I |  |  | I |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  | 1 |  |  |  |  |  |  |  |
| 7 |  |  | 1 |  |  |  |  |  | 1 |  |  |  |
| 8 |  |  |  |  |  |  | 1 |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  | 1 |  |  |  |  |
| 10 |  |  |  |  |  | 1 |  |  | 1 |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  | 1 |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  | 1 |  |

Для выделения маршрутов можно использовать матрицу достижимости, которая также представляет собой для заданного графа управления квадратную таблицу размером 12x12.

Номер столбца матрицы определяет номер вершины графа, из которого возможно достичь другие вершины этого же графа, используя цикличные и ацикличные маршруты. Номера строк определяют номера достижимых вершин графа управления. Из вершины 1 возможно достичь вершины со 2-й по 12-ю, т. е. достижимы все вершины графа (см. рис. 2.37). Для первого столбца матрицы достижимости строки, начиная со второй, заполняются единицами (табл. 2.10). Далее заполняются столбцы для остальных вершин аналогично.

Таблица 2.10. Матрица достижимости для задачи простого числа

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  | 1 |  |  | 1 |  |  |  |  |  |  |
| 3 | 1 | 1 | 1 |  |  | 1 |  |  |  |  |  |  |
| 4 | 1 |  |  | | | 1 |  | 1 |  | 1 |  |  |
| 5 | 1 |  |  | < яя | | 1 |  |  |  | 1 |  |  |
| 6 | 1 |  |  | 1 |  | | |  |  | 1 | 1 |  |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 1 |  |
| 8 | 1 |  |  |  |  | 1 |  | Ж |  | 1 | 1 |  |
| 9 | 1 | 1 |  |  | 1 | 1 | 1 | 1 | ш | 1 | 1 |  |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  | 1 |  |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 12 | 1 | 1 |  |  |  | 1 |  | 1 |  | 1 |  |  |

Выделенные диагональные элементы матрицы определяют номера вершин, которые входят в состав циклических маршрутов. Идентичные строки матрицы определяют номера вершин, которые входят в состав ацикличных маршрутов.

Элементы, выделенные черным цветом, соответствуют маршруту т4: 4-5-6, элементы, выделенные штриховкой, соответствуют маршруту т5: 7-8-9. Первые два «серых» элемента, «черные» элементы и последние два «серых» элемента соответствуют m3: 2-3-4—5-6-10-11.

Все строки матрицы получились идентичными, так как все вершины графа входят в состав ацикличных маршрутов:

ml: 1-2—3—4—5-6-10-11—12; m2: 1-2-3-7-8-9-10-11-12.

Проведем оценку алгоритмической сложности по третьему критерию. В соответствии с третьим критерием необходимо выделить все реально возможные маршруты управления:

ml: 1-2-3-4-5-6-10-11-12; рх = 3;

m2: 1-2-3-7-8-9-10—И—12; р2 = 3;

m3: 1-2-3-4-5-6-4—5-6-10-11-12; р2 = 4;

m4: 1—2—3—7—8—9—7—8—9—10-11-12; р4 = 4;

т5: 1-2-3-4-5-6-4—5-6-10-11-2-3-4-5-6-4-5-6-10-П-12; р5 = 8;

тб: 1 -2—3—7—8—9-7—8-9-10-11—2-3—7—8—9-7-8-9—10-П-12; рй = 8;

т7: 1-2-3-4-5-6-4—5-6-10-11-2-3-7-8-9-7-8-9-10-11-12; р7 = 8;

т8: 1-2-3-7-8-9-7-8-9-10-11-2-3-4-5-6-4-5-6-10-П-12; = 8.

Оценку структурной сложности программы рассчитываем по следующему соотношению:

Sj = р^ + р-> + рх + /?4 + /?5 + + р2 + =3 + 3 + 4 + 4 + 8 + 8 + 8 + 8 = 46.

Исходя из полученных результатов расчета метрик структурной сложности по первому (X] = 8), второму (S2 = 11) и третьему (53 = 46) критериям выделения маршрутов можно сделать вывод, что программа, характеризуемая заданным графом управления, имеет невысокую алгоритмическую сложность, так как количество используемых в тексте операторов условий равно 5.

Для тестирования программы с данным управляющим графом необходимо проверить от 8 до 46 тестовых маршрутов обработки исходных данных.

1. Задача «Сортировка массива»

Необходимо провести сортировку значений элементов одномерного массива по возрастанию.

По разработанному алгоритму построить управляющий граф и оценить алгоритмическую сложность возможной программы.

Реализация решения

Алгоритм решения задачи представлен на рис. 2.38.

В соответствии с разработанным алгоритмом граф потока управления будет иметь вид, показанный на рис. 2.39.

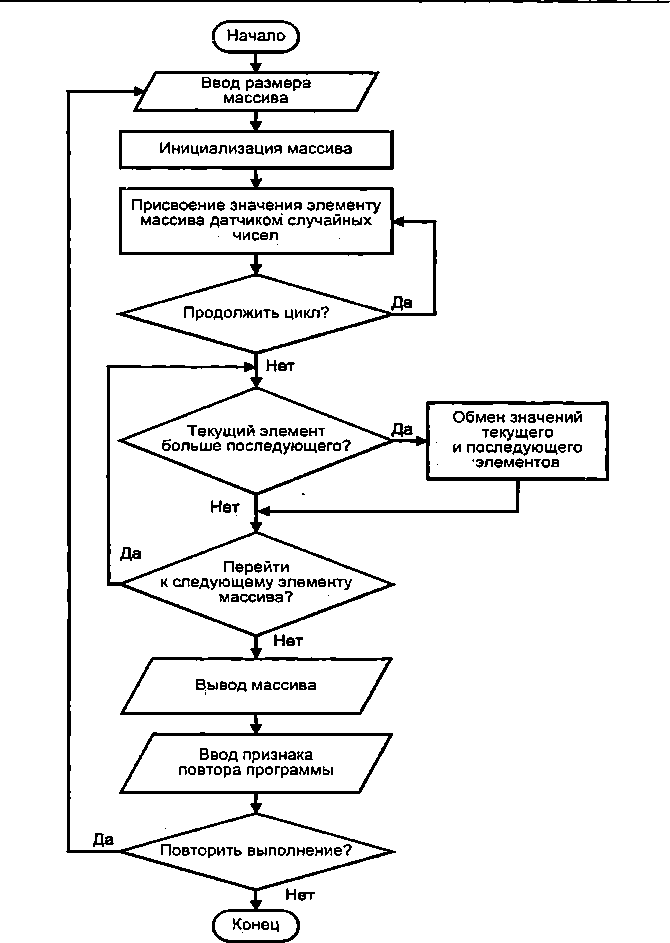


Рис. 2.38. Алгоритм решения задачи сортировки массива

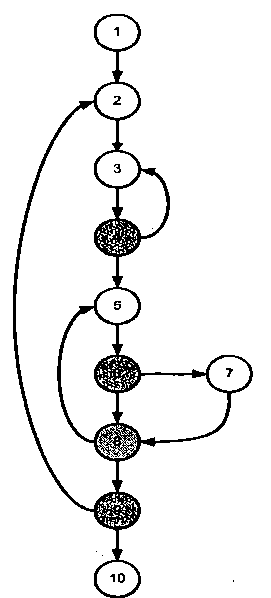


Рис. 2.39. Управляющий граф задачи сортировки массива

Оценка алгоритмической сложности

Проведем оценку алгоритмической сложности графа по первому критерию, для чего определим минимальный набор маршрутов, проходящих через каждый оператор ветвления и по каждой дуге.

Возможный маршрут управления:

ml: 1-2-3-4-3-4-5-6-7-8-5-6-8-9-2-3-4-5-6-8-9-10; Р\ = 11.

Анализ выделенного маршрута показывает, что он проходит по всем вершинам ветвления и хотя бы один раз по каждой дуге представленного графа. Минимальное количество маршрутов для заданного графа составляет 1, так как этим маршрутом охватываются все дуги и вершины графа. Тонированные вершины обозначают операторы ветвления, в перечне пунктов маршрута номера этих вершин выделены шрифтом с Подчеркиванием/Количество таких пунктов маршрута - 11. Таким образом, в соответствии с первым критерием алгоритмической сложности количество маршрутов 1, количество вершин ветвления в маршруте 11:

Ji=p, = ll.

Проведем оценку алгоритмической сложности по второму критерию. Необходимо определить число проверок каждого линейно независимого цикла и линейно независимого ациклического участка программы. Количество проверок определяется цикломатическим числом графа, которое определяется следующим соотношением:

Z = /7„ + 1,

где пв - число вершин ветвления.

Число вершин ветвления в заданном графе составляет 4, следовательно,

Z = /)„ + 1 = 4 + 1 = 5.

Таким образом, общее число циклических и ациклических маршрутов составляет 5. Выделим эти маршруты на заданном графе:

* ациклические маршруты:

ml: 1 -2-3-4-5-6-7-8-9-10-11 -12-13; рх= 4; m2: 1-2-3-4-5-6-7-9-10-11-12-13; р2 = 4;

* циклические маршруты:

m3: 3-4;р3= 1;

/л4: 5—6—8; р4 = 2;

/т/5: 2-3-4-5-6-7-8-9; р5= 2.

Тестирование указанных маршрутов позволяет проверить все операторы ветвления программы. Метрика структурной сложности определяется по следующему соотношению:

S2 =Р\ + Pi^ Рз + Р4 Т Ръ + />б = 4 + 4 + 1 + 2 + 4= 15.

Для организации автоматического анализа заданного графа по второму критерию с помощью вычислительных средств построим матрицы смежности и достижимости.

Матрица смежности представляет собой квадратную матрицу, размер которой определяется количеством вершин графа. Заданный граф имеет 10 вершин, следовательно, матрица смежности будет иметь размер Ю^Ю. При заполнении матрицы следует придерживаться следующего правила: для дуги, соединяющей вершину 1с вершиной 2 (см. рис. 2.39) в первый столбец и вторую строку матрицы смежности записываем 1 (табл. 2.11).

Таблица 2.11. Матрица смежности

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  | 1 |  |
| 3 |  | 1 |  | 1 |  |  |  |  |  |  |
| 4 |  |  | 1 ■ |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  | 1 |  |  |
| 6 |  |  |  |  | 1 |  |  |  |  |  |
| 7 |  |  |  |  |  | 1 |  |  |  |  |
| 8 |  |  |  |  |  | 1 | 1 |  |  |  |
| 9 |  |  |  |  |  |  |  | 1 |  |  |
| 10 |  |  |  |  |  |  |  |  | 1 |  |

Номер столбца соответствует номеру вершины, из которой выходит дуга, номер строки соответствует номеру вершины, в которую входит дуга. Таким же образом заполняем матрицу для всех дуг управляющего графа.

Для выделения маршрутов можно использовать матрицу достижимости, которая представляет собой для заданного графа управления квадратную таблицу размером 10x10. Номер столбца матрицы определяет номер вершины графа, из которого возможно достичь другие вершины этого же графа, используя цикличные и ацикличные маршруты. Номера строк определяют номера достижимых вершин графа управления. Из вершины с 1 возможно достичь вершины со 2-й по 10-ю, т. е. достижимы все вершины графа. Для первого столбца матрицы достижимости строки, начиная со второй, заполняются единицами (табл. 2.12). Далее заполняются столбцы для остальных вершин аналогично.

Выделенные диагональные элементы матрицы определяют номера столбцов-вершин, которые входят в состав циклических маршрутов. Идентичные строки матрицы определяют номера столбцов- вершин, которые входят в состав ацикличных маршрутов.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 |  | ш |  | 1 | 1 |  | 1 | 1 |  |
| 4 | 1 |  |  |  | 1 | 1 | 1 |  |  |  |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | W//A | 1 | 1 |  |
| 8 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 |
| 10 |  |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Элементы, выделенные штриховкой в левую сторону, соответствуют маршруту m3, элементы, выделенные штриховкой в правую сторону, - маршруту от4. Элементы, помеченные единицами полужирного шрифта белого и черного цвета, соответствуют маршруту т5. Элементы, помеченные единицами белого цвета, соответствуют тб. Все строки матрицы получились идентичными, так как все вершины графа входят в состав ацикличных маршрутов ml и m2.

Проведем оценку алгоритмической сложности по третьему критерию. В соответствии с третьим критерием необходимо выделить все реально возможные маршруты управления:

ml: 1-2-3-4-5-6-7-8-9-10; рх = 4; m2: 1-2-3-4-5-6-8-9-Т0; р2 = 4; m3: 1-2-3-4-3-4-5-6-7-8-5-6-8-9-10; р3= 6; т4: 1 -2-3 -4-5 -6-8-9-2-3-4-5-6-8-9-10; />4 = 8.

Оценку структурной сложности программы рассчитываем по следующему соотношению:

S3=pi + р2+ +р4 = 4 + 4 + 6 + 8 = 22.

Исходя из полученных результатов расчета метрик структурной сложности по первому (S) = 11), второму (S2 = 20) и третьему (S) = 41) критериям выделения маршрутов можно сделать вывод, что программа, характеризуемая заданным графом управления, имеет невысокую алгоритмическую сложность, так как количество используемых в тексте операторов условий равно 4, для проверки которых необходимо провести от 11 до 22 тестовых вариантов исходных данных.

1. Задача «Поиск максимального числа»

Необходимо определить максимальное из натуральных чисел, не превышающих К, которое нацело делится на М. Числа К и М вводятся с клавиатуры по запросу программы. По разработанному алгоритму построить граф потока управления и оценить алгоритмическую сложность программы.

Реализация решения

Алгоритм решения задачи представлен на рис. 2.40.

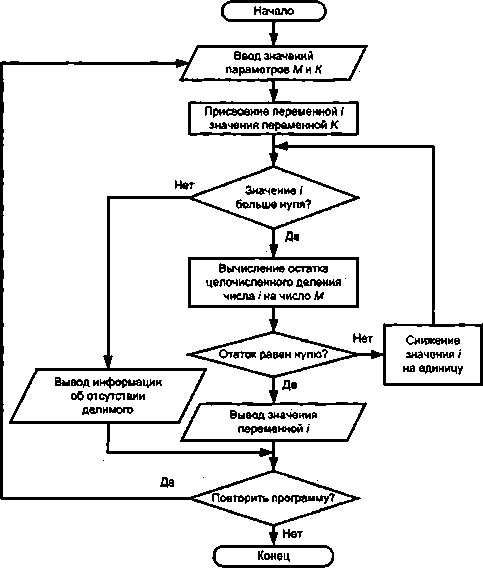


Рис. 2.40. Алгоритм решения задачи о поиске максимального числа

Граф потока управления для программы поиска максимального числа приведен на рис. 2.41.

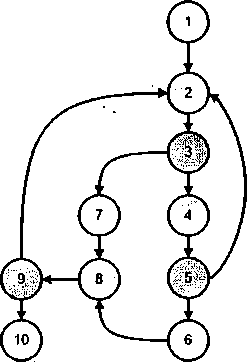


Рис. 2.41. Граф потока управления для задачи поиска максимального числа

Оценка алгоритмической сложности

Проведем оценку алгоритмической сложности графа по первому критерию. Определим минимальный набор маршрутов, проходящих через каждый оператор ветвления и по каждой дуге.

Возможный маршрут управления:

т\: 1-2-3-4-5-2-3-4-5-6-8-9-2-3-7-8-9-10; р, = 6.

Выделенный маршрут проходит по всем вершинам ветвления и хотя бы один раз по каждой дуге представленного графа. Минимальное количество маршрутов для заданного графа составляет 1, так как этим маршрутом охватываются все дуги и вершины графа. Тонировка вершин обозначает операторы ветвления. В перечне маршрута номера этих вершин выделены полужирным шрифтом с подчеркиванием. Если пересчитать полужирные номера выделенного маршрута, то их количество составляет 6. Таким образом, в соответствии с первым критерием алгоритмической сложности количество маршрутов 1, количество вершин ветвления в маршруте 6: S\ =p\ = 6.

Проведем оценку алгоритмической сложности по второму критерию. Необходимо определить число проверок каждого линейно независимого цикла и линейно независимого ациклического участка программы. Количество проверок определяется цикломатическим числом графа, которое определяется следующим соотношением:

Z=h„+ 1,

где па - число вершин ветвления.

Число вершин ветвления в заданном графе составляет 3, следовательно, цикломатическое число определяется следующим образом:

Z = w„ + 1 = 3 + 1 = 4.

Таким образом, общее число циклических и ациклических маршрутов составляет 4. Выделим эти маршруты на заданном графе:

* ациклические маршруты:

ml: 1-2-3-4-5-8-9-10; рх = 3;

m2:1-2-3-7-8-9-10; р2 = 2;

* циклические маршруты:

m3: 2-3-4-5;р2 = 2;

/и4: 2—3—7—8—9; р4 = 2. «

Тестирование указанных маршрутов позволяет проверить все операторы ветвления программы. Метрика структурной сложности определяется по следующему соотношению:

\*$2= Р\ + Pif Рз+7,4 = 3 + 5 + 2 + 2=12.

Для организации автоматического анализа заданного графа по второму критерию с помощью вычислительных средств построим матрицы смежности и достижимости.

Матрица смежности представляет собой квадратную матрицу, размер которой определяется количеством вершин графа. Заданный граф имеет 10 вершин, следовательно, матрица смежности будет иметь размер 10x10. При заполнении матрицы следует придерживаться следующего правила: для дуги, соединяющей вершину 1 с вершиной 2, в первый столбец и вторую строку матрицы смежности записываем 1 (табл. 2.13). Номер столбца соответствует номеру вершины, из которой выходит дуга, номер строки соответствует номеру вершины, в которую входит дуга. Таким же образом заполняем матрицу для всех дуг управляющего графа.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  | 1 |  |  |  | 1 |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |
| 4 |  |  | 1 |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |  |
| 6 |  |  |  |  | Т |  |  |  |  |  |
| 7 |  |  | 1 |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  | 1 | 1 |  |  |  |
| 9 |  |  |  |  |  |  |  | 1 |  |  |
| 10 |  |  |  |  |  |  |  |  | 1 |  |

Для выделения маршрутов можно использовать матрицу достижимости, которая представляет собой для заданного графа управления квадратную таблицу размером 10x10. Номер столбца матрицы определяет номер вершины графа, из которого можно достичь другие вершины этого же графа, используя цикличные и ацикличные маршруты. Номера строк определяют номера достижимых вершин графа управления. Из вершины с 1 возможно достичь вершины со 2-й по 10-ю, т. е. достижимы все вершины графа (рис. 2.41.). Для первого столбца матрицы достижимости строки, начиная со второй, заполняются единицами. Аналогично заполняются столбцы для остальных вершин (табл. 2.14).

Таблица 2.14. Матрица достижимости задачи поиска максимального числа

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 | / | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 3 | 1 | 1 | / | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 4 | 1 | 1 | 1 |  | | 1 | 1 | 1 | 1 |  |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 6 | 1 | 1 | 1 | 1 | 1 | — | 1 | I | 1 |  |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | — | 1 |  |  |
| 8 |  | 1 | 1 | 1 |  | 1 | 1 |  | 1 |  |
| 9 | 1 | 1 | 1 |  | 1 | 1 |  | 1 | — |  |
| 10 | 1 | 1 |  |  | 1 | 1 | 1 | 1 | 1 |  |

Выделенные диагональные элементы матрицы определяют номера столбцов-вершин, которые входят в состав циклических маршру

тов. Идентичные строки матрицы определяют номера столбцов- вершин, которые входят в состав ацикличных маршрутов.

Элементы матрицы достижимости, выделенные черным цветом, соответствуют маршруту m3. Элементы, помеченные единицами полужирным курсивом белого и черного цвета, соответствуют маршруту т4. Все строки матрицы получились идентичными, так как все вершины графа входят в состав ацикличных маршрутов ml и m2.

Проведем оценку алгоритмической сложности по третьему критерию. В соответствии с третьим критерием необходимо выделить все реально возможные маршруты управления:

ml: 1-2-3-4—5-6-8-9-10; р, = 3;

m2: 1-2-3-7-8-9-10; р2 = 2;

m3: 1-2-3-4-5-2-3-4-5-6-8-9-10; р3 = 5;

т4: 1-2-3-7-8-9-2-3-4-5-2-5-4-5-6-8-9-10; рл= 7;

т5: 1-2-3-4-5-2-3-4-5-6-8-9-2-3-7-8-2-Ю; р5 = 7

тб: 1-2-3-7-8-9-2-3-7-8-9-10; р6 = 4.

Оценку структурной сложности программы рассчитываем по следующему соотношению:

53 =Р\ + р2 + рз + Р4 + Ps + Рь = 3 + 2 + 5 + 7 + 7 + 4 = 28. t

Исходя из полученных результатов расчета метрик структурной сложности по первому (Si = 6), второму (S2 = 9) и третьему (S3 = 28) критериям выделения маршрутов можно сделать вывод, что программа, характеризуемая заданным графом управления, имеет низкую алгоритмическую сложность, так как количество используемых в тексте операторов условий 3, для проверки которых необходимо проверить от 6 до 28 тестовых вариантов исходных данных.

1. Задачи для самостоятельного решения
2. Задачи с разработкой программы

В задачах 1-10, предлагаемых для самостоятельного решения, необходимо выполнить следующее с целью оценки алгоритмической сложности:

* разработать алгоритм решения задачи;
* построить граф потока управления:
* сформировать маршруты тестирования в соответствии с критериями 1, 2 и 3;
* определить значение цикломатического числа, характеризующего структурную сложность программ;
* сформировать матрицы смежности и достижимости;
* провести анализ полученных результатов, сформировав содержательные выводы.

Задача 1. Вывести на экран все натуральные числа из диапазона от А до В, сумма цифр которых равна S. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В и заданная сумма цифр S вводятся с клавиатуры.

Задача 2. Вывести на экран все натуральные трехзначные и пятизначные числа из диапазона от А до В, значение которых кратно 13. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В вводятся с клавиатуры.

Задача 3. Вывести на экран все натуральные числа из диапазона от А до В, у которых совпадают старшая и младшая цифры. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В вводятся с клавиатуры.

Задача 4. Вывести на экран для всех натуральных чисел из диапазона от А до В сами числа и суммы цифр, находящихся на нечетных позициях. Номера позиций отсчитываются с единицы, начиная с младшей цифры. Границы диапазона А и Б вводятся с клавиатуры.

Задача 5. Вывести на экран для всех натуральных чисел из диапазона от А до В сами числа и суммы трех последних (младших) цифр. Границы диапазона А и В вводятся с клавиатуры. Если диапазон не содержит трехзначных или чисел с большей разрядностью, сформировать сообщение «Диапазон указан ошибочно».

Задача 6. Вычислить и вывести на экран суммы К старших (находящихся слева) цифр натурального числа А. Число А и значение К вводятся с клавиатуры. Если количество цифр в числе меньше К, сформировать сообщение «Значение числа К слишком велико».

Задача 7. Вывести на экран все натуральные шестизначные числа из диапазона от А до В, у которых совпадают сумма трех младших и трех старших цифр. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В вводятся с клавиатуры.

Задача 8. Вывести на экран все натуральные числа из диапазона от А до В, в записи которых цифра 7 встречается ровно N раз. При отсутствии чисел с указанными свойствами выдать на экран сообщение «Требуемых чисел нет». Границы диапазона А и В и значение N ввести с клавиатуры.

Задача 9. Вывести на экран сумму S ряда чисел:

с 2 3 4 5

5 = — + — + —+ —+•••

3 7 11 15

Сумма вычисляется до получения слагаемого, меньшего заданного значения А. Значение А вводится с клавиатуры.

Задача 10. Вычислить суммы S первых N слагаемых последовательности чисел, образуемых по правилу:

С 1 2 3 4

5 = -+ — + —+ —+•••.

13 5 7

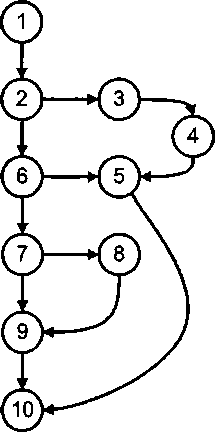
Сумму вычислить двумя способами: Si - суммирование от первого слагаемого до N-го слагаемого, S2 - суммирование от N-го слагаемого до первого слагаемого. Значение N ввести с клавиатуры. Вывести на экран вычисленные суммы S] и S2, а также значение модуля разности между ними.

1. Задачи по оценке алгоритмической сложности на основе управляющих графов

Исходя из предложенного графа потока управления, разработанного на основе алгоритма и программы, в задачах 11 - 20, предлагаемых для самостоятельного решения, необходимо выполнить следующее с целью оценки алгоритмической сложности:

* сформировать маршруты тестирования в соответствии с критериями 1, 2 и 3;
* определить значение цикломатического числа, характеризующего структурную сложность программ;
* сформировать матрицы смежности и достижимости;
* провести анализ полученных результатов, сформировав содержательные выводы.

Задача 11. Структура управляющего графа представлена на рис. 2.42.



Задача 12. Структура управляющего графа представлена на рис. 2.43.

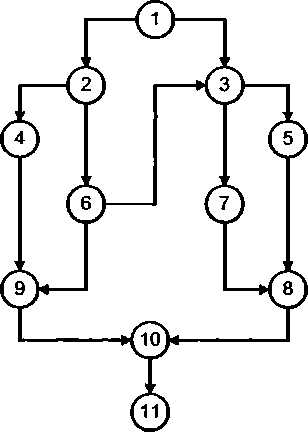


Рис. 2.42. Структура графа потока р,1С 243 Стру1аура афа потока

управления к задаче 11 управления к задаче 12

Задача 13. Структура управляющего графа представлена на рис. 2.44. Задача 14. Структура управляющего графа представлена на рис. 2.45. Задача 15. Структура управляющего графа представлена на рис. 2.46. Задача 16. Структура управляющего графа представлена на рис. 2.47. Задача 17. Структура управляющего графа представлена на рис. 2.48. Задача 18. Структура управляющего графа представлена на рис. 2.49. Задача 19. Структура управляющего графа представлена на рис. 2.50. Задача 20. Структура управляющего графа представлена на рис. 2.51.

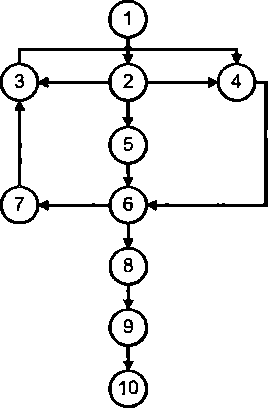


Рис. 2.44. Структура графа потока управления к задаче 13

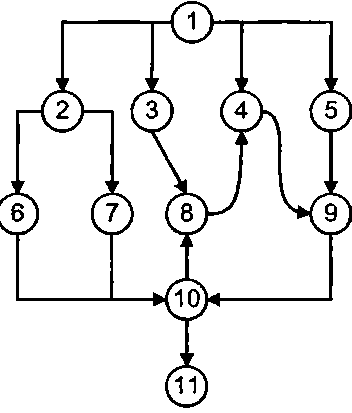


Рис. 2.46. Структура графа потока Рис. 2.47. Структура графа потока

управления к задаче 15 управления к задаче 16

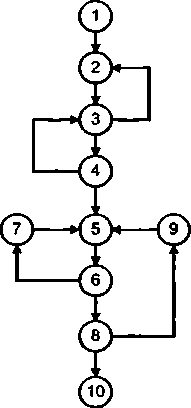
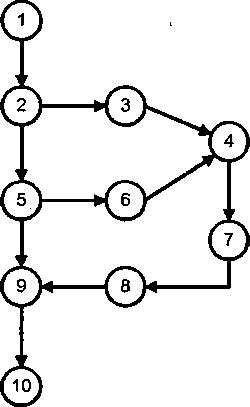


Рис. 2.45. Структура графа потока управления к задаче 14



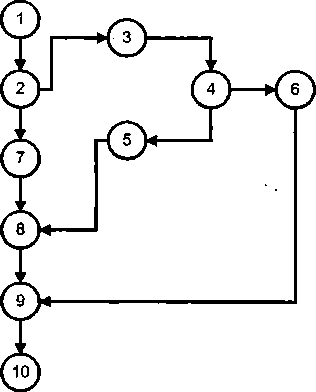


Рис. 2.48. Структура графа потока

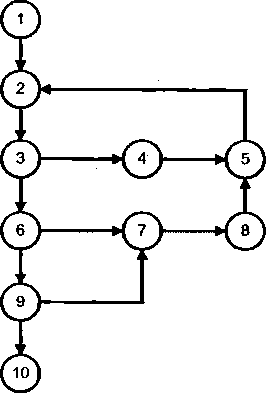


Рис. 2.49. Структура графа потока управления к задаче 18

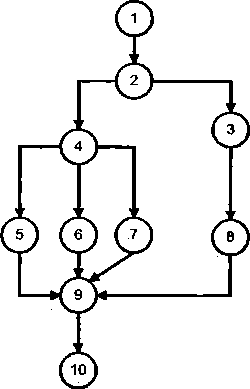


Рис. 2.51. Структура графа потока управления к задаче 20

управления к задаче 17

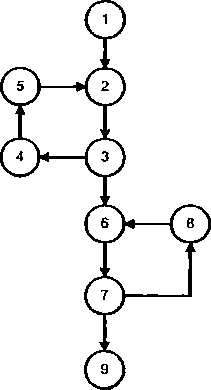


Рис. 2.50. Структура графа потока управления к задаче 19