

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Основы программирования

по направлению подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) подготовки:

DevOps-инженерия в администрировании инфраструктуры ИТ-разработки

Форма обучения

Очная

Квалификация

Бакалавр

Год приема

2024

СОГЛАСОВАНО:
Руководитель ОП
С.П.Сущенко

Председатель УМК
С.П.Сущенко

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.

ОПК-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения.

ОПК-5 Способен устанавливать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства.

ОПК-6 Способен использовать в педагогической деятельности научные основы знаний в сфере информационно-коммуникационных технологий.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем

ИОПК-2.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий

ИОПК-2.3 Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности

ИОПК-3.3 Использует современные информационные технологии, в том числе отечественного производства на всех этапах разработки программных систем

ИОПК-5.1 Определяет порядок и особенности процесса инсталляции программного и аппаратного обеспечения для информационных и автоматизированных систем

ИОПК-5.2 Инсталлирует программное и аппаратное обеспечение

ИОПК-5.3 Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности

ИОПК-6.1 Владеет прикладным программным обеспечением для разработки методической документации для педагогической деятельности

ИОПК-6.2 Способен автоматизировать задачи в области педагогической деятельности, разрабатывать и сопровождать информационные системы для поддержки педагогической деятельности

ИОПК-6.3 Использует системные знания в области информационно-коммуникационных технологий для организации педагогической деятельности

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- письменные контрольные работы в течении каждого из двух семестров (по 4 работы в каждом семестре);

- лабораторные работы (реализация на компьютере заданий в виде программ на языке Паскаль в 1-м семестре и на языке Си во 2-м семестре)

Вопросы и задания, включаемые в контрольные работы, проверяющие достижение закрепленных за дисциплиной компетенций по следующим индикаторам: ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИОПК-3.3, ИОПК-5.1, ИОПК-5.2, ИОПК-5.3, ИОПК-6.1, ИОПК-6.2 и ИОПК-6.3.

1. Дан целочисленный массив A из n элементов. Алгоритм должен в одном цикле найти сумму минимального и максимального значений в массиве A, а также номера элементов с этими значениями. Доказать правильность алгоритма методом инварианта.

$$S = x - \frac{x^2}{2 \cdot 2!} + \frac{x^3}{3 \cdot 3!} - \frac{x^4}{4 \cdot 4!} + \dots \pm \frac{x^n}{n \cdot n!} + \dots$$

2. Задан ряд ϵ s. Написать рекуррентную формулу для вычисления членов суммы, а также для суммы S. Алгоритм должен вычислить сумму S для заданного x. Доказать правильность алгоритма методом инварианта. Какова его трудоемкость, и почему?

3. Рекуррентная последовательность Герона и алгоритм вычисления квадратного корня. Какова его трудоемкость, и почему?

4. Дан целочисленный массив X из n элементов. Алгоритм должен найти максимальную сумму S двух рядом расположенных элементов в массиве X, а также номера элементов с этими значениями. Доказать правильность алгоритма методом инварианта.

$$S = 1 - \frac{1}{x \cdot 2!} + \frac{1}{x^2 \cdot 3!} - \frac{1}{x^3 \cdot 4!} + \dots \pm \frac{1}{x^{n-1} \cdot n!} + \dots$$

5. Задан ряд ϵ s. Написать рекуррентную формулу для вычисления членов суммы, а также для суммы S. Алгоритм должен вычислить сумму S для заданного x. Доказать правильность алгоритма методом инварианта. Какова его трудоемкость, и почему?

6. Алгоритм вычисления корня функции $y = f(x)$ методом дихотомии. Какова его трудоемкость, и почему?

7. Дан целочисленный массив B из n элементов. Алгоритм должен в одном цикле найти номер j элемента с наименьшим и номер k элемента с наибольшим значением по абсолютной величине в массиве B. Доказать правильность алгоритма методом инварианта.

$$S = \frac{x^2}{2 \cdot 2} - \frac{x^4}{3 \cdot 2^2} + \frac{x^6}{4 \cdot 2^3} - \dots \pm \frac{x^{2n}}{(n+1) \cdot 2^n} \dots$$

8. Задан ряд ϵ s. Написать рекуррентную формулу для вычисления членов суммы, а также для суммы S. Алгоритм должен вычислить сумму S для заданного x. Доказать правильность алгоритма методом инварианта. Какова его трудоемкость, и почему?

9. Алгоритм вычисления цифр целого числа P ($P \geq 0$) при заданном значении основания q. Какова его трудоемкость и почему?

10. Дан целочисленный массив A из n элементов и два числа, b и g ($b \leq g$). Алгоритм должен в одном цикле найти сумму значений в массиве A, меньших b и сумму значений в массиве A, больших g. Доказать правильность алгоритма методом инварианта.

$$S = x^2 - \frac{x^4}{2!} + \frac{x^6}{3!} - \frac{x^8}{4!} + \dots \pm \frac{x^{2n}}{n!} + \dots$$

11. Задан ряд ϵ s. Написать рекуррентную формулу для вычисления членов суммы, а также для суммы S. Алгоритм должен вычислить сумму S для заданного x. Доказать правильность алгоритма методом инварианта. Какова его трудоемкость, и почему?

12. Алгоритм вычисления наибольшего общего делителя двух чисел. Доказать его правильность. Какова его трудоемкость и почему?

13. Алгоритм сортировки вставками. Доказательство правильности, трудоемкости (в худшем и в лучшем случаях).

14. Рекурсивный алгоритм поиска элемента в упорядоченном массиве из n чисел методом дихотомии. Доказательство правильности, трудоемкости, глубины рекурсии.

15. Какова трудоемкость алгоритма слияния для двух упорядоченных массивов, и почему?

16. Алгоритм сортировки методом выбора максимального элемента. Доказательство правильности, трудоемкости.

17. Алгоритм решения игры «Ханойские башни». Доказательство правильности, трудоемкости. Доказать, что он решает задачу за минимальное число перекладываний дисков.

18. Какова трудоемкость алгоритма поиска элемента в упорядоченном массиве из n чисел методом дихотомии, и почему?

19. Алгоритм сортировки массива методом слияния (использовать, как готовую процедуру, алгоритм слияния двух последовательностей в одну). Доказательство правильности, трудоемкости, глубины рекурсии.

20. Алгоритм удаления повторяющихся элементов в упорядоченном массиве. Доказательство правильности, трудоемкости.

21. Какова трудоемкость алгоритма обменной сортировки (с двойным циклом) в худшем и в лучшем случае, и почему?

22. Алгоритм слияния двух упорядоченных массивов в один. Доказательство правильности, трудоемкости.

2.3 Алгоритм косвенной сортировки с использованием индексного массива на основе алгоритма обменной сортировки. Доказательство правильности, трудоемкости.

24. Какова трудоемкость алгоритма сортировки слиянием, и почему?

25. Алгоритм поиска элемента в упорядоченном массиве из n чисел методом дихотомии. Доказательство правильности, трудоемкости.

26. Алгоритм пузырьковой сортировки. Доказательство правильности, трудоемкости.

27. Какова трудоемкость алгоритма сортировки методом выбора в худшем и в лучшем случае, и почему?

28. Алгоритм (процедура) генерации всех таких размещений из n чисел по m элементов ($m < n$), чтобы сумма чисел в каждом из них была равна заданной величине Q . Вызов из программы. Доказательство правильности, трудоемкости, глубины рекурсии.

29. Написать программу, которая: 1) вводит n , 2) вводит n чисел и формирует из них список, 3) упорядочивает список методом пузырька, 4) выводит результат.

30. Алгоритм (процедура) генерации всех таких расстановок ферзей на шахматной доске $n \times n$, чтобы они не «били» друг друга. (Ферзи бьют по вертикалям, горизонталям и диагоналям.) Вызов из программы. Доказательство правильности, трудоемкости, глубины рекурсии.

31. Написать программу, которая: 1) вводит n , 2) вводит n чисел и формирует из них список, 3) ищет в списке максимальное и минимальное значение, 4) выводит результат.

32. Алгоритм (процедура) генерации всех таких перестановок из n чисел, чтобы на 1-м месте всегда стояло число 1 или 2, а сумма первых k чисел в каждой перестановке была равна заданной величине Q . Вызов из программы. Доказательство правильности, трудоемкости, глубины рекурсии.

33. Написать программу, которая: 1) вводит n_1 , 2) вводит n_1 чисел и формирует из них 1-й список, 3) вводит n_2 , 4) вводит n_2 чисел и формирует из них 2-й список, 5) в предположении, что оба списка упорядочены, сливает их в общий упорядоченный список, 6) выводит результат.

34. Алгоритм (процедура) генерации всех таких сочетаний из n чисел по m элементов, чтобы сумма чисел в каждом сочетании была равна заданной величине Q . Вызов из программы. Доказательство правильности, трудоемкости, глубины рекурсии.

35. Написать программу, которая: 1) вводит n , 2) вводит n чисел и формирует из них список, 3) упорядочивает список (путем вызова процедуры), 4) выводит результат. Требуется также написать процедуру сортировки слиянием списка.

36. Алгоритм сравнения двух строк с учетом перекодировки символов (с отождествлением больших и малых букв) в виде функции. Как строится массив перекодировки? Трудоемкость алгоритма?

37. Множество представлено в виде массива A номеров элементов универсума. Элементы массива упорядочены. Алгоритм вставки в множество еще одного элемента так, чтобы массив A оставался упорядоченным. Трудоемкость?

38. Задана таблица результатов соревнований по троеборью в виде: (фамилия, вид 1, вид 2, вид 3). Строки таблицы перемешаны. Алгоритм вычисления мест по отдельным видам и общего места по сумме мест отдельных видов. Трудоемкость?

39. Алгоритм поиска всех совпадений подстроки d (длиной m) с подряд идущими символами в символьном массиве S длиной n . (Символы произвольные!) Трудоемкость алгоритма?

40. Множество представлено в виде массива A номеров элементов универсума. Элементы массива неупорядочены. Алгоритм поиска элемента номер k в таком множестве. Трудоемкость?

41. Заданы 3 таблицы результатов соревнований по троеборью в виде: (фамилия, вид 1), (фамилия, вид 2), (фамилия, вид 3). В каждой из таблиц строки перемешаны. Алгоритм получения общей таблицы в виде: (фамилия, вид 1, вид 2, вид 3), в которую включены только те спортсмены, кто имеет результаты по всем видам. Трудоемкость?

42. Алгоритм формирования массива всех слов, имеющих в тексте. Слова состоят только из букв, отделяются друг от друга символами-разделителями (пробелами и др.). Массив слов должен быть упорядоченным, и в нем не должно быть повторений. Каким образом определяется в алгоритме, является ли символ разделителем? Трудоемкость алгоритма?

43. Множество представлено в виде (бинарного) логического массива A (количество элементов универсума равно n). Алгоритмы: 1) поиска элемента номер k в таком множестве; 2) добавления элемента номер k в такое множество; 3) удаления элемента номер k из множества. Трудоемкость?

44. Задана таблица оценок учащихся в виде: (фамилия, предмет, оценка). У каждого из учащихся имеется по несколько оценок по каждому из 2-х предметов (математика и физика). В таблице строки перемешаны. Алгоритм вычисления средних оценок по каждому предмету в отдельности для каждого из учащихся. Трудоемкость?

45. Написать программу на C, которая выполняет следующие действия: ввод 2-х имен файлов, открытие этих файлов, ввод n , выделяет динамически память для двух массивов целых из n элементов, вводит из 1-го файла n чисел (в 1-й массив), упорядочивает их вызовом функции, выполняющей сортировку слиянием, в конце выводит результат во 2-й файл. Написать также функцию сортировки слиянием для массива целых чисел. Трудоемкость программы?

46. Написать программу на C, которая выполняет следующие действия: ввод 2-х имен файлов, открытие этих файлов, ввод n , вводит из 1-го файла n чисел и формирует из них список, затем упорядочивает его вызовом функции сортировки списков, выводит результат во 2-й файл. Написать функцию сортировки слиянием списков, в которой вызывается функция слияния двух списков в один (функцию слияния не нужно описывать). Трудоемкость программы?

47. Написать программу на C, которая выполняет следующие действия: ввод строки символов S (содержащей слово из букв), затем ввод имени файла, вызов функции обработки файла (с параметрами – именем файла и строку символов S), вывод результата. Функция должна открыть файл, посимвольно ввести из файла текст, выделить из текста в процессе ввода отдельные слова, содержащие только буквы, и подсчет результата: сколько раз слово S встретилось в файле. Трудоемкость программы?

48. Написать программу на C, которая выполняет следующие действия: ввод имени файла, ввод n , открывает файл для вывода и вызывает функцию генерации перестановок n ферзей, не бьющих друг друга на доске $n \times n$ причём каждую перестановку выводит в файл, как отдельную строку. Написать функцию генерации перестановок ферзей и их вывода в файл. Трудоемкость программы?

49. Алгоритм (в виде функции) быстрого возведения в большую целочисленную степень p квадратной матрицы ($n \times n$), включая функции вычисления произведения двух матриц, копирования матриц и т.п. Доказательство корректности. Трудоемкость алгоритма (и почему)? Написать программу на Си с вводом данных, вызовом функции и выводом результата.

50. Какова трудоемкость алгоритма вычисления обратной матрицы (и почему)?

51. Алгоритм (в виде функции) решения системы n уравнений с n неизвестными. Коэффициенты левой части заданы квадратной матрицей A , правая часть – вектором B . Задана точность решения ϵ . Предусмотреть случай вырожденной матрицы. Трудоемкость алгоритма (и почему)? Написать программу на Си с вводом данных, вызовом функции и выводом результата.

52. Какова трудоемкость алгоритма вычисления определителя квадратной матрицы (и почему)?

53. Алгоритм (в виде функции) вычисления определителя квадратной матрицы A . Задана точность решения ϵ . Предусмотреть случай вырожденной матрицы. Трудоемкость алгоритма (и почему)? Написать программу на Си с вводом данных, вызовом функции и выводом результата.

54. Какова трудоемкость алгоритма решения системы уравнений в общем случае (m уравнений и n неизвестных), и почему?

55. Алгоритм (в виде функции) вычисления обратной матрицы D , если задана квадратная матрица A . Задана точность решения ϵ . Предусмотреть случай вырожденной матрицы. Трудоемкость алгоритма (и почему)? Написать программу на Си с вводом данных, вызовом функции и выводом результата.

56. Какова трудоемкость решения системы n уравнений с n неизвестными (и почему)?

57. Написать на C++ программу, которая вводит данные взвешенного графа (число вершин n , матрицу расстояний M , вводит номер начальной вершины a , вычисляет в массиве R кратчайшее расстояние от вершины a до всех остальных, выводит результат.

Вывести формулу трудоемкости этого алгоритма для графа из n вершин и m ребер.

58. Написать на C++ программу, которая вводит данные неориентированного графа (число вершин n , число ребер m , ребра, как пары номеров вершин), формирует матрицу смежности M , вычисляет в массиве S принадлежность вершин компонентам связности, а также в массиве R вычисляет номера вершин по порядку просмотра (вызовом функции просмотра графа вглубь в цикле), выводит результаты (массивы S и R). Описать также функцию просмотра графа вглубь, которая вызывается.

Вывести формулу трудоемкости этого алгоритма для графа из n вершин и m ребер.

59. Написать на C++ программу, которая вводит данные ориентированного графа (число вершин n , число ребер m , ребра, как пары номеров вершин), формирует матрицу смежности M , вычисляет в массиве P топологическую упорядоченность вершин, проверяет, существует ли эта упорядоченность, выводит результат (массив P).

Вывести формулу трудоемкости этого алгоритма для графа из n вершин и m ребер.

60. Написать на C++ программу, которая вводит данные неориентированного графа (число вершин n , число ребер m , ребра, как пары номеров вершин), формирует массив смежных вершин (массивы S , L и D), вычисляет в массиве S принадлежность вершин компонентам связности, а также в массиве R вычисляет номера вершин по порядку просмотра (вызовом функции просмотра графа вглубь в цикле), выводит результаты (массивы S и R). Описать также функцию просмотра графа вглубь.

Вывести формулу трудоемкости этого алгоритма для графа из n вершин и m ребер.

61. Написать на C++ программу, которая:

- 1) вводит: n – число вершин, m – число ребер, m пар чисел, каждая пара задает ребро ориентированного графа;
- 2) формирует из них массив смежных вершин (массивы S , L по n элементов и массив D из m элементов);
- 3) вычисляет эйлеров цикл в виде списка (предполагается, что он заведомо существует);
- 4) выводит результат в виде последовательности вершин, через которые проходит цикл.

Какова трудоемкость этого алгоритма для графа из n вершин и m ребер и почему?

62. Что такое жизненный цикл программы? Основные этапы жизненного цикла.

63. Написать на C++ программу, которая:

- 1) вводит: n – число вершин, m – число ребер, m пар чисел, каждая пара задает ребро неориентированного графа;
- 2) формирует из них массив смежных вершин (массивы S , L по n элементов и массив D из m элементов);
- 3) вычисляет первый попавшийся гамильтонов цикл (остальные циклы не ищет!);
- 4) выводит последовательность вершин – цикл (или сообщение, что цикла не существует).

Какова трудоемкость этого алгоритма для графа из n вершин и m ребер и почему?

64. Чем программный комплекс отличается от простой программы и насколько сложнее его создавать?

65. Написать на C++ программу, которая:

- 1) вводит: n – число вершин, m – число ребер, m пар чисел, каждая пара задает ребро ориентированного графа;
- 2) формирует из них матрицу смежности (двумерный массив M);
- 3) вычисляет первый попавшийся гамильтонов путь (остальные пути не ищет!);
- 4) выводит последовательность вершин – путь (или сообщение, что пути не существует).

Какова трудоемкость этого алгоритма для графа из n вершин и m ребер и почему?

66. Что такое программный продукт? Насколько сложнее его создавать по сравнению с простой программой?

67. Написать на C++ программу, которая:

- 1) вводит: n – число вершин, m – число ребер, m пар чисел, каждая пара задает ребро неориентированного графа;
- 2) формирует из них матрицу смежности (двумерный массив M);
- 3) вычисляет эйлеров цикл в виде списка (предполагается, что он заведомо существует);
- 4) выводит результат в виде последовательности вершин, через которые проходит цикл.

Какова трудоемкость этого алгоритма для графа из n вершин и m ребер и почему?

68. Какая документация необходима для программы, и для кого она предназначена? Что должна содержать документация?

В каждую контрольную работу включаются 2-3 вопроса (задания).

Результаты контрольной работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если даны правильные ответы на все теоретические вопросы и все задачи решены без ошибок.

Оценка «хорошо» выставляется, если даны в целом правильные ответы (с незначительными ошибками) на все теоретические вопросы и все задачи решены без существенных ошибок.

Оценка «удовлетворительно» выставляется, если даны в целом правильные ответы (с существенными ошибками) на часть теоретических вопросов и часть задач решены без существенных ошибок.

Оценка «неудовлетворительно» выставляется, если не даны правильные на теоретические вопросы и задачи решены с существенными ошибками или вообще не решены.

Оценки за каждую контрольную работу приводятся в балльно-рейтинговую форму, до 25 баллов за оценку «отлично». Таким образом, за 4 контрольных работы в каждом семестре можно получить до 100 баллов.

Текущий контроль по дисциплине основан на применении 200-балльной шкалы оценивания в каждом семестре. Проводится оценивание выполнения контрольных работ (по 100-балльной шкале) и лабораторных заданий (по 100-балльной шкале). Критерии оценивания публикуются в методических материалах к дисциплине. Результаты текущего контроля определяются по общей сумме баллов и фиксируются в форме контрольной точки не менее одного раза в семестр. Обучающиеся, набравшие не менее 35 баллов, выполнившие не менее одной контрольной работы и не менее одного обязательного задания, получают аттестацию. Обучающиеся, не выполнившие хотя бы одно из перечисленных выше требований, считаются не аттестованными.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Оценка за промежуточную аттестацию в каждом семестре вычисляется на основе суммы баллов по четырем письменным контрольным работам и сданным лабораторным работам, проверяющих достижение закрепленных за дисциплиной компетенций по следующим индикаторам: ОПК-2, ОПК-5, ОПК-6, ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИОПК-3.3, ИОПК-5.1, ИОПК-5.2, ИОПК-5.3, ИОПК-6.1, ИОПК-6.2 и ИОПК-6.3. Таблица перевода оценок из 200-балльной шкалы в 5-балльную:

Баллы -> оценки (итог)		
От	До	
173		отлично
112	172	хорошо
67	111	удовлетворительно
0	66	неудовлетворительно

Условие получения удовлетворительной оценки – выполнение всех контрольных работ и обязательное выполнение 4-5 определенных заданий в семестре.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Задания.

1. Дан целочисленный массив X из n элементов. Алгоритм должен найти максимальную сумму S двух рядом расположенных элементов в массиве X , а также

номера элементов с этими значениями. Доказать правильность алгоритма методом инварианта.

2. Задан ряд
$$S = 1 - \frac{1}{x \cdot 2!} + \frac{1}{x^2 \cdot 3!} - \frac{1}{x^3 \cdot 4!} + \dots \pm \frac{1}{x^{n-1} \cdot n!} + \dots$$
, значение x и точность

вычисления ряда ϵ . Написать рекуррентную формулу для вычисления членов суммы, а также для суммы S . Доказать правильность алгоритма методом инварианта.

3. Рекурсивный алгоритм сортировки массива методом слияния. Доказательство правильности, трудоемкости, глубины рекурсии.

4. Алгоритм поиска элемента в упорядоченном массиве из n чисел методом дихотомии. Доказательство правильности, трудоемкости.

5. Рекурсивный алгоритм генерации всех возможных перестановок n чисел. Доказательство правильности, трудоемкости.

6. Написать программу, которая: 1) вводит n , 2) вводит n чисел и формирует из них список, 3) ищет в списке максимальное и минимальное значение, 4) выводит результат.

7. Два множества представлены в виде массивов A и B номеров элементов универсума. Элементы массивов упорядочены. Алгоритм вычисления пересечения этих множеств.

8. Два множества представлены в виде массивов A и B номеров элементов универсума. Элементы массивов упорядочены. Алгоритм вычисления объединения этих множеств.

9. Два множества представлены в виде битовых массивов A и B для универсума из n элементов. Алгоритм вычисления пересечения этих множеств.

10. Два множества представлены в виде битовых массивов A и B для универсума из n элементов. Алгоритм вычисления объединения этих множеств.

11. Алгоритм поиска всех совпадений подстроки d (длиной m) с подряд идущими символами в символьном массиве S длиной n . (Символы произвольные!) Трудоемкость алгоритма.

12. Алгоритм быстрого возведения в большую целочисленную степень p квадратной матрицы ($n \times n$). Доказательство корректности. Трудоемкость алгоритма.

13. Алгоритм решения системы n уравнений с n неизвестными. Задана точность решения ϵ . Предусмотреть случай вырожденной матрицы. Трудоемкость алгоритма.

14. Дана матрица расстояний взвешенного графа и номер A начальной вершины. Алгоритм (Дейкстры) должен вычислять в массиве R кратчайшее расстояние от вершины A до всех остальных вершин. Трудоемкость алгоритма.

15. Дана симметричная матрица расстояний взвешенного графа. Алгоритм (Прима) должен вычислять рёбра минимального остова этого графа. Трудоемкость алгоритма.

16. Дана матрица смежности неориентированного графа. Алгоритм должен вычислять компоненты связности этого графа. Трудоемкость алгоритма.

17. Дана матрица смежности ориентированного графа и номер A начальной вершины. Алгоритм должен вычислять кратчайшее расстояние от вершины A до всех остальных вершин. Трудоемкость алгоритма.

18. Дана матрица смежности ориентированного графа. Алгоритм должен вычислять первый попавшийся гамильтонов цикл (остальные циклы не ищет). Трудоемкость алгоритма.

19. Дана матрица смежности ориентированного графа. Алгоритм должен проверить, существует ли цикл Эйлера. Трудоемкость алгоритма.

Вопросы.

1. Как доказать правильность алгоритма, содержащего цикл, методом инварианта?
2. Как вычислить трудоёмкость алгоритма, содержащего циклы?
3. Как любой алгоритм упорядочения чисел превратить в алгоритм косвенного упорядочения (сортировки)?

4. Как построить алгоритм перебора всех возможных вариантов (алгоритм бэктрекинга)?
5. Как представляется в программах множество в виде битового (логического) массива?
6. Как представляется в программах множество в виде массива номеров элементов?
7. Как сравниваются две строки символов в лексикографическом порядке?
8. Что такое таблица перекодировки и как она используется для сравнения символов?
9. Как квадратная матрица чисел приводится к треугольному виду по методу Гаусса-Жордана?
10. Как представляется в программе неориентированный и ориентированный граф в виде матрицы смежности?
11. Как представляется в программе неориентированный и ориентированный граф в виде списков смежных вершин?
12. Как представляется в программе взвешенный граф в виде матрицы?
13. Как разрабатывается большая программа по методу сверху-вниз?
14. Как разрабатывается большая программа по методу снизу-вверх?
15. Что такое жизненный цикл программы?
16. Что такое программный продукт?
17. Какая документация необходима для программы, и для кого она предназначена?

Ответ должен содержать формальную постановку задач, ее решение и интерпретацию полученных выводов.

Информация о разработчиках

Ю.Л. Костюк, д-р техн. наук, профессор, кафедра теоретических основ информатики ТГУ, профессор.

И.Л. Фукс, кафедра теоретических основ информатики ТГУ, старший преподаватель.