

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет
имени Т.Г. Шевченко»

Рыбницкий филиал

Кафедра информатики и программной инженерии

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ**

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Б1.В.ДВ.5.1 «Введение в алгоритмы»

Код наименование дисциплины

Основной образовательной программы высшего образования по направлению
подготовки 09.03.04 «Программная инженерия»

индекс наименование направления

профиль Разработка программно-информационных систем

наименование профиля подготовки

квалификация выпускника бакалавр

форма обучения заочная

Разработчик

преподаватель Нагаевская Н.В.

(ФИО, должность)

Обсужден на заседании кафедры

«24» сентябре 2020 г.

Протокол № 2

Зав. кафедрой ИиПИ, доцент

Л.А. Тягульская Тягульская Л.А.

Рыбница 2020 г.

ПАСПОРТ
фонда оценочных средств по учебной дисциплине

Введение в алгоритмы
(наименование дисциплины)

1. В результате изучения дисциплины «Введение в алгоритмы» обучающийся должен:

1.1. Знать:

- основные понятия теории алгоритмов: интуитивная концепция алгоритма, уточнения понятия алгоритма (машины Тьюринга и нормальные алгоритмы Маркова), понятия вычислимости, разрешимости;
- основные неразрешимые массовые проблемы;
- временные и емкостные оценки алгоритмов.

1.2. Уметь:

- составлять программы машин Тьюринга и схемы нормальных алгоритмов для решения простых вычислительных задач;
- проводить анализ алгоритмов.

1.3. Владеть:

- навыками разработки и отладки программ на алгоритмических языках программирования, а также навыками анализа алгоритмов.

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины и их наименование *	Код контролируемой компетенции (или ее части)	Наименование оценочного средства**
1	Раздел 1. Введение	ПК-1, ПК-2, ПК-21	Комплект тестов Комплект заданий для контрольной работы
2	Раздел 2. Алгоритмы и вычислимость	ПК-1, ПК-2, ПК-21	Комплект тестов Комплект заданий для контрольной работы
3	Раздел 3. Анализ алгоритмов	ПК-1, ПК-2, ПК-21	Комплект тестов Комплект заданий для контрольной работы
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства**
1		ПК-1, ПК-2, ПК-21	Комплект КИМ

«УТВЕРЖДАЮ»
зав. кафедрой ИиПИ,
доцент _____ Л. А. Тягульская
«___» _____ 2020 г.

**Итоговый тест по дисциплине
«Введение в алгоритмы»
для студентов V курса, з/о
направления «Программная инженерия» профиля подготовки
«Разработка программно-информационных систем»**

Образец итоговых тестовых заданий по дисциплине «Введение в алгоритмы»

Номер задания	Наименование темы задания
ДЕ 1,2. Алгоритмы и вычислимость (критерий освоения ДЕ: не менее 15 правильно выполненных заданий)	
1-10	Алгоритм. Понятие и свойства алгоритма.
11-20	Машина Тьюринга
21-27	Нормальные алгоритмы Маркова
28-40	Рекурсивные алгоритмы
ДЕ 3. Анализ алгоритмов (критерий освоения ДЕ: не менее 15 правильно выполненных заданий)	
1-5	Сравнительная оценка алгоритмов
6-24	Трудоемкость основных алгоритмических конструкций
25-33	Сложностные классы задач
34-40	Метод декомпозиции

ДЕ-1,2

- Алгоритм-это:
 - Указание на выполнение действий,
 - Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи,
 - Процесс выполнения вычислений, приводящих к решению задачи
- Свойство алгоритма – дискретность, выражает, что:
 - Команды должны следовать последовательно друг за другом,
 - Каждая команда должна быть описана в расчете на конкретного исполнителя,
 - Разбиение алгоритма на конечное число команд
- Формальное исполнение алгоритма – это:
 - Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,
 - Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,
 - Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем автоматически

- d) Исполнение алгоритма осуществляется исполнителем на уровне его знаний
4. Укажите основные черты алгоритмов
- Дискретность
 - Массовость
 - Непрерывность
 - Ассоциативность
 - Детерминированность
 - Индивидуальность
5. Какой алгоритм называется линейным:
- Выполнение операций зависит от условия,
 - Операции выполняются друг за другом,
 - Одни и те же операции выполняются многократно
 - Присутствие всех возможных операций в одном алгоритме
6. Свойство алгоритма записываться в виде упорядоченной совокупности отделенных друг от друга предписаний (директив):
- понятность
 - определенность
 - дискретность
 - массовость
7. Графическое задание алгоритма – это:
- Способ представления алгоритма с помощью геометрических фигур,
 - Представление алгоритма в форме таблиц и расчетных формул,
 - Система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения
8. Как называется конструкция блок-схемы, изображенная на рисунке:
- выполнение операций
 - начало-конец алгоритма
 - вызов вспомогательного алгоритма
 - ввод/вывод данных
- 
9. В расчете на кого должен строиться алгоритм:
- В расчете на ЭВМ,
 - В расчете на умственные способности товарища,
 - В расчете на конкретного исполнителя
10. Циклическим называется алгоритм, в котором:
- Выполнение операций зависит от условия,
 - Операции выполняются друг за другом,
 - Одни и те же операции выполняются многократно
11. Машина Тьюринга является:
- терминальной вычислительной машиной
 - абстрактной вычислительной машиной
 - комплексной вычислительной машиной
12. Любой конечный упорядоченный набор символов из известного алфавита носит название:
- строка
 - слово

с) модуль

13. Результат применения команды машины Тьюринга $q_11 \rightarrow q_2a_0Л$ к машинному слову $a_01 * q_11a_0$

- а) $a_01 * q_21a_0$
- б) $a_01 * 1q_2a_0$

- с) $a_01q_2 * a_0$
- д) $a_01 * q_11a_0$

14. Результат применения команды машины Тьюринга $q_11 \rightarrow q_2a_0П$ к машинному слову $a_01 * q_11a_0$

- а) $a_01 * a_0q_1a_0$
- б) $a_01 * a_0q_2a_0$

- с) $a_01q_2 * a_0$
- д) $a_01q_1 * a_0$

15. Машина Тьюринга задана функциональной схемой

$$\begin{aligned} q_10 &\rightarrow q_11П \\ q_1a_0 &\rightarrow q_00С \end{aligned}$$

В какое слово переработает машина слово $\alpha = 01110$, исходя из стандартного начального положения

- а) 0110
- б) 011110

- с) 01111
- д) 011100

16. Установите соответствие

- а) пустая буква
- б) начальное состояние
- с) стоп-состояние

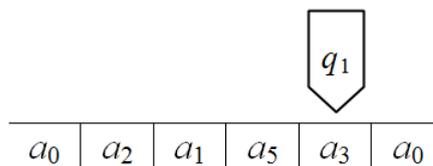
- 1) q_1
- 2) q_0
- 3) a_1
- 4) a_0

17. В машине Тьюринга предписание L для лентопротяжного механизма означает

- а) переместить ленту вправо
- б) переместить ленту влево

- с) остановить машину
- д) занести в ячейку символ

18. Конфигурация (машинное слово), соответствующая изображению



а) $a_0a_2a_1a_5a_3a_0$

с) $a_0a_2a_1a_5a_3qa_0$

б) $a_0a_2a_1a_5q_1a_3a_0$

д) $q_1a_0a_2a_1a_5a_3a_0$

19. Если значение вычислимой по Тьюрингу функции $f(x_1, \dots, x_n)$ не определено, то ...

- а) Машина останавливается через конечное число шагов, на ленте записано пустое слово
- б) Машина останавливается через конечное число шагов, на ленте записано исходное слово
- с) Машина останавливается через конечное число шагов, на ленте записано слово «error»
- д) Машина работает бесконечно

20. Функция $f(x)$ является вычислимой по Тьюрингу. Машина Тьюринга, вычисляющая ее значения, задана функциональной схемой

$$\begin{aligned} q_10 &\rightarrow q_10Л, & q_1a_0 &\rightarrow q_2a_0П, & q_11 &\rightarrow q_30П, \\ q_20 &\rightarrow q_00П, & q_30 &\rightarrow q_0a_0Л \end{aligned}$$

Формульное выражение функции $f(x)$ имеет вид

а) $f(x) = x$

с) $f(x) = x + 2$

б) $f(x) = x + 1$

д) $f(x) = x \div 1$

21. Результат марковской подстановки $21 \rightarrow 3$ в слово 521421

- a) 52143
- b) 5343
- c) 531421
- d) 533433

22. Марковская подстановка $21 \rightarrow 3$ не применима к словам

- a) 521421
- b) 5241
- c) 21
- d) 12

23. Результат марковской подстановки $21 \rightarrow \Lambda$ в слово 521421

- a) 54
- b) 21521421
- c) 5421
- d) 52142121

24. Процесс работы нормального алгоритма считается завершенным, если на данном шаге

- a) Применена последняя формула в списке формул марковских подстановок, задающих данный алгоритм
- b) Применена заключительная формула подстановки
- c) Ни одна подстановка схемы не подходит
- d) Понятно, что процесс подстановок не сможет остановиться

25. Нормальный алгоритм не применим к исходным данным, если на данном шаге

- a) Применена последняя формула в списке формул марковских подстановок, задающих данный алгоритм
- b) Применена заключительная формула подстановки
- c) Ни одна подстановка схемы не подходит
- d) Понятно, что процесс подстановок не сможет остановиться

26. Результат применения нормального алгоритма $\alpha\beta \rightarrow \Lambda$, $\beta\alpha \rightarrow \alpha\beta$ к слову $R = \beta\beta\alpha\alpha\beta\alpha\beta$

- a) Алгоритм не применим к этому слову
- b) β
- c) α
- d) $\beta\beta$
- e) $\alpha\alpha$

27. Результат применения нормального алгоритма $ab \rightarrow bd$, $db \rightarrow ba$, $bba \rightarrow abb$, $c \rightarrow \Lambda$ к слову $R = abbc$

- a) Алгоритм не применим к этому слову
- b) bb
- c) aa
- d) cc

28. Значение функции $\lambda(4)$ равно ... (где $\lambda(x)$ – оператор сдвига)

29. Значение функции $\lambda(O(6))$ равно ... (где $\lambda(x)$ – оператор сдвига, $O(x)$ – оператор обнуления)

30. Значение функции $O(\lambda(7))$ равно ... (где $\lambda(x)$ – оператор сдвига, $O(x)$ – оператор обнуления)

31. Значение функции $I_3^4(3,2,1,7)$ равно ... (где $I_m^n(x_1, x_2, \dots, x_n)$ – оператор выбора)

32. Значение функции $I_2^5(3,2,1,7,3)$ равно ... (где $I_m^n(x_1, x_2, \dots, x_n)$ – оператор выбора)

33. Схема примитивной рекурсии задает функцию

$$F(x, y) = \begin{cases} g(x) = x \\ h(x, y, z) = \lambda(z) \end{cases}$$

- a) Усеченная разность $F(x, y) = x \dot{-} y$
- b) Сложение чисел $F(x, y) = x + y$
- c) Произведение чисел $F(x, y) = x \cdot y$
- d) Возведение в степень $F(x, y) = x^y$

34. Схема примитивной рекурсии задает функцию

$$F(x, y) = \begin{cases} g(x) = 0 \\ h(x, y, z) = x + z \end{cases}$$

- a) Усеченная разность $F(x, y) = x \dot{-} y$
- b) Сложение чисел $F(x, y) = x + y$
- c) Произведение чисел $F(x, y) = x \cdot y$
- d) Возведение в степень $F(x, y) = x^y$

35. Функция $F(x, y) = (x \dot{-} y) + (y \dot{-} x)$ является

- a) $F(x, y) = \max(x, y)$
- b) $F(x, y) = \min(x, y)$
- c) $F(x, y) = |x - y|$
- d) $F(x, y) = \text{НОД}(x, y)$

36. Функция, которая может быть получена из простейших функций с помощью конечного числа применений операторов суперпозиции, примитивной рекурсии и минимизации, называется

- a) Примитивно рекурсивной функцией
- b) Частично рекурсивной функцией
- c) Всюду рекурсивной функцией
- d) Сложно рекурсивной функцией

37. Установите соответствие. Функция $f(x)$ задана с помощью операции

- | | |
|---|-------------------------|
| a) $f(x) = I_1^3(x, y, z)$ | 1) минимизации |
| b) $f(x) = \text{Rec}(g, h(x, y))$ | 2) примитивной рекурсии |
| c) $f(x) = \mu y [g(x, y) = 0]$ | 3) суперпозиции |
| d) $f(x) = \text{Sub}(\varphi(y_1, y_2); f_1(x), f_2(x))$ | |

38. Значение функции $F(4,2)$, заданной схемой примитивной рекурсии, равно

$$F(x, y) = \begin{cases} g(x) = x \\ h(x, y, z) = \lambda(z) \end{cases}$$

39. Значение функции $F(4,2)$, заданной схемой примитивной рекурсии, равно

$$F(x, y) = \begin{cases} g(x) = 0 \\ h(x, y, z) = x + z \end{cases}$$

40. Значение функции $F(2,5)$ равно ...

$$F(x, y) = (x \dot{-} y) + (y \dot{-} x)$$

ДЕ-3

1. Установите соответствие между сложностью алгоритма и типом зависимости

- | | |
|--------------------------|---------------------|
| a) $O(n)$ | 1) Логарифмическая |
| b) $O(n \cdot \log_2 n)$ | 2) Линейная |
| c) $O(n^2)$ | 3) Экспоненциальная |
| d) $O(2^n)$ | 4) Полиномиальная |

2. Вычислить размер входных значений, которые можно обработать за 1 с, если единица времени = 1мс и сложность алгоритма $O(2^n)$

3. Вычислить размер входных значений, которые можно обработать за 1 м, если единица времени = 1мс и сложность алгоритма $O(n^3)$

4. Два алгоритма A1 и A2 , решающих одну и ту же задачу размерности $n=10^6$. A1 имеет сложность $O_1(n^2)$ и выполняется на суперкомпьютере с быстродействием 10 8оп/с; A2 имеет сложность $O_2(n \cdot \log_2 n)$ и выполняется на обычном компьютере с быстродействием 10⁶ оп/с. Найти время решения задачи $t_1 - ?$, $t_2 - ?$

5. В работах по теории сложности алгоритм называется *полиномиальным в среднем*, если для входов длины n и времени работы алгоритма T , выполняется:

a) $\exists \varepsilon > 0 \mathbf{E}_n T^\varepsilon = O(n)$

b) $\forall C \mathbf{E}_n T = O(n^C)$

c) $\forall \varepsilon > 0 \mathbf{E}_n T^\varepsilon = O(n)$

d) $\exists C \mathbf{E}_n T = O(n^C)$

6. Трудоемкость алгоритма – это

a) количество времени, потраченное для решения конкретной задачи

b) количество «элементарных» операций совершаемых алгоритмом для решения конкретной задачи

c) количество денег, потраченное для решения конкретной задачи

7. Временная сложность алгоритма – это время, необходимое для выполнения алгоритма, зависящее от ...

a) исходных данных

b) вычислительной техники

c) исполнителя

8. Установите алгоритмы в порядке возрастания сложности

a) Экспоненциальные алгоритмы

b) Полиномиальные алгоритмы

c) Линейные алгоритмы

9. Количественно-зависимые по трудоемкости алгоритмы это

a) алгоритмы, трудоемкость которых определяется не размерностью входа, а конкретными значениями обрабатываемых слов памяти

b) алгоритмы, трудоемкость которых зависит как от количества данных на входе, так и от значений входных данных

c) алгоритмы, функция трудоемкости которых зависит только от размерности конкретного входа, и не зависит от конкретных значений

10. Параметрически-зависимые по трудоемкости алгоритмы это

a) алгоритмы, трудоемкость которых определяется не размерностью входа, а конкретными значениями обрабатываемых слов памяти

b) алгоритмы, трудоемкость которых зависит как от количества данных на входе, так и от значений входных данных

с) алгоритмы, функция трудоемкости которых зависит только от размерности конкретного входа, и не зависит от конкретных значений

11. Количественно-параметрические по трудоемкости алгоритмы это

а) алгоритмы, трудоемкость которых определяется не размерностью входа, а конкретными значениями обрабатываемых слов памяти

б) алгоритмы, трудоемкость которых зависит как от количества данных на входе, так и от значений входных данных

с) алгоритмы, функция трудоемкости которых зависит только от размерности конкретного входа, и не зависит от конкретных значений

12. Алгоритмы поиска минимума и максимума в массиве относятся к

а) параметрически-зависимым по трудоемкости

б) количественно-параметрическим по трудоемкости

с) количественно-зависимым по трудоемкости

13. Алгоритмы умножения матриц, умножение матрицы на вектор относятся к

а) параметрически-зависимым по трудоемкости

б) количественно-параметрическим по трудоемкости

с) количественно-зависимым по трудоемкости

14. Алгоритмы вычисления стандартных функций с заданной точностью путем вычисления соответствующих степенных рядов относятся к

а) параметрически-зависимым по трудоемкости

б) количественно-параметрическим по трудоемкости

с) количественно-зависимым по трудоемкости

15. Установите соответствия между обозначениями анализа алгоритмов по трудоемкости

а) $F_a^{\wedge}(N)$

б) $F_a^{\vee}(N)$

с) $\bar{F}_a(N)$

1) среднее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N

2) наибольшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N

3) наименьшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N

16. Что из ниже перечисленного является «элементарными» операциями:

а) Простое присваивание: $a = b$

б) Одномерная индексация $a[i]$: (адрес $(a) + i \cdot \text{длина элемента}$)

с) Арифметические операции: $(*, /, -, +)$

д) Вычисление факториала: $a!$

е) Логические операции $(l1) \{or, and, not\} (l2)$

ф) Вычисление логарифма: $\log_n k$

17. Что из ниже перечисленного является «элементарными» конструкциями

- a) конструкция «Следования»
- b) конструкции «Ветвление»
- c) конструкция «Массив»
- d) конструкция «Цикл»

18. Что из ниже перечисленного не является методикой перехода к временным оценкам

- a) Пооперационный анализ
- b) Метод Гиббсона
- c) Метод декомпозиции
- d) Метод прямого определения среднего времени

19. Способ композиции нормальных алгоритмов будет суперпозицией, если:

- a) выходное слово первого алгоритма является входным для второго;
- b) существует алгоритм C , преобразующий любое слово p , содержащееся в пересечении областей определения алгоритмов A и B ;
- c) алгоритм D будет суперпозицией трех алгоритмов ABC , причем область определения D является пересечением областей определения алгоритмов A , B и C , а для любого слова p из этого пересечения $D(p) = A(p)$, если $C(p) = e$, $D(p) = B(p)$, если $C(p) = e$, где e — пустая строка;
- d) существует алгоритм C , являющийся суперпозицией алгоритмов A и B такой, что для любого входного слова p $C(p)$ получается в результате последовательного многократного применения алгоритма A до тех пор, пока не получится слово, преобразуемое алгоритмом B .

20. Способ композиции нормальных алгоритмов будет объединением, если:

- a) выходное слово первого алгоритма является входным для второго;
- b) существует алгоритм C , преобразующий любое слово p , содержащееся в пересечении областей определения алгоритмов A и B ;
- c) алгоритм D будет суперпозицией трех алгоритмов ABC , причем область определения D является пересечением областей определения алгоритмов A , B и C , а для любого слова p из этого пересечения $D(p) = A(p)$, если $C(p) = e$, $D(p) = B(p)$, если $C(p) = e$, где e — пустая строка;
- d) существует алгоритм C , являющийся суперпозицией алгоритмов A и B такой, что для любого входного слова p $C(p)$ получается в результате последовательного многократного применения алгоритма A до тех пор, пока не получится слово, преобразуемое алгоритмом B .

21. Способ композиции нормальных алгоритмов будет разветвлением, если:

- a) выходное слово первого алгоритма является входным для второго;
- b) существует алгоритм C , преобразующий любое слово p , содержащееся в пересечении областей определения алгоритмов A и B ;
- c) алгоритм D будет суперпозицией трех алгоритмов ABC , причем область определения D является пересечением областей определения алгоритмов A , B и C , а для любого слова p из этого пересечения $D(p) = A(p)$, если $C(p) = e$, $D(p) = B(p)$, если $C(p) = e$, где e — пустая строка;
- d) существует алгоритм C , являющийся суперпозицией алгоритмов A и B , такой, что для любого входного слова p $C(p)$ получается в результате последовательного многократного применения алгоритма A до тех пор, пока не получится слово, преобразуемое алгоритмом B .

22. Способ композиции нормальных алгоритмов будет итерацией, если:

- a) выходное слово первого алгоритма является входным для второго;

- b) существует алгоритм C , преобразующий любое слово p , содержащееся в пересечении областей определения алгоритмов A и B ;
- c) алгоритм D будет суперпозицией трех алгоритмов ABC , причем область определения D является пересечением областей определения алгоритмов A B и C , а для любого слова p из этого пересечения $D(p) = A(p)$, если $C(p) = e$, $D(p) = B(p)$, если $C(p) = e$, где e — пустая строка;
- d) существует алгоритм C , являющийся суперпозицией алгоритмов A и B , такой, что для любого входного слова p $C(p)$ получается в результате последовательного многократного применения алгоритма A до тех пор, пока не получится слово, преобразуемое алгоритмом B .

23. Трудоемкость какой конструкции вычисляется по формуле $F = f_1 + \dots + f_k$

- a) конструкция «Следования»
- b) конструкции «Ветвление»
- c) конструкция «Массив»
- d) конструкция «Цикл»

24. Что из ниже перечисленного не является методикой перехода к временным оценкам

- a) Пооперационный анализ
- b) Метод Гиббсона
- c) Метод декомпозиции
- d) Метод прямого определения среднего времени

25. К какому классу задач относятся полиномиальные алгоритмы?

- a) P-класс
- b) NP-класс
- c) E-класс

26. К какому классу задач относятся экспоненциальные алгоритмы?

- a) P-класс
- b) NP-класс
- c) E-класс

27. Какой метод предполагает проведение совокупного анализа по трудоемкости и переход к временным оценкам на основе принадлежности решаемой задачи к одному из стандартных типов?

- a) Пооперационный анализ
- b) Метод Гиббсона
- c) Метод прямого определения среднего времени

28. Какие задачи являются практически разрешимыми

- a) задачи P-класса
- b) задачи NP-класса
- c) задачи NP-полного класса

29. Какие из ниже приведенных задач являются NP

- a) алгоритм комплексного умножения
- b) задача о коммивояжере
- c) выбор кратчайшего пути распространения информации по всем узлам компьютерной сети

30. Задачи, которые можно быстро решить, относятся к...

- a) P-класс
- b) NP-класс

31. Задачи, которые можно быстро проверить, относятся к...
- a) P-класс
 - b) NP-класс
32. Какой класс задач является наиболее сложным
- a) P-класс
 - b) NP-класс
 - c) NP-полный класс
33. К недерменированным алгоритмам относятся :
- a) задачи P-класса
 - b) задачи NP-класса
 - c) задачи NP-полного класса
34. Метод декомпозиции это
- a) структурное разбиение системы
 - b) объединение частей в целое
 - c) структурное разбиение и объединение в целое
35. Причиной для использования метода декомпозиции является...
- a) легче изучить частные проблемы, чем всю проблему в целом
 - b) возможность разделить работу между отдельными исполнителями
36. Что не является методом декомпозиции
- a) рассмотрение проблемы в рамках отдельных интервалов времени
 - b) объединение сходных экономических субъектов и объектов в максимально большие группы
 - c) разбиение на основе научных дисциплин
 - d) разбиение в соответствие с интересами
 - e) разбиение применительно к различным географическим областям
37. Для того, чтобы формальная модель стала основанием для декомпозиции, ее следует...
- a) наполнить содержанием
 - b) разбить на части
 - c) объединить из ее частей
38. Для представления иерархической структуры нельзя использовать
- a) дерево
 - b) график
 - c) список
39. Расположите в правильном порядке алгоритм метода декомпозиции
- a) Выбор формальной модели ИС
 - b) Определение объекта анализа
 - c) Декомпозиция по компонентам модели ИС
 - d) Проверка на элементарность
 - e) Остановка декомпозиции
 - f) Определение исследуемой системы ИС
40. Основными требованиями, предъявляемыми, к декомпозиции являются
- a) иерархичность структуры
 - b) массовость
 - c) возможность использования количественных показателей

d) целостность представления анализируемого объекта

Ключи к тесту

ДЕ 1	
1	b
2	c
3	c
4	a,b,e
5	b
6	c
7	a
8	d
9	c
10	c
11	b
12	b
13	c
14	b
15	b
16	a-4, b-1, c-2
17	b
18	b
19	d
20	d
21	c
22	b,d
23	c
24	b,c
25	d
26	b
27	a
28	5
29	1
30	0
31	1
32	2
33	b
34	c
35	c
36	b
37	b-2, c-1, d-3
38	6
39	8
40	3
ДЕ-2	
1	a-2, b-1, c-4, d-3
2	9
3	39

4	28ч, 20с
5	a
6	b
7	a
8	c,b,a
9	c
10	a
11	c
12	b
13	c
14	a
15	a-2, b-3, c-4
16	a,b,c,e
17	a,b,d
18	c
19	a
20	b
21	c
22	d
23	a
24	c
25	a
26	c
27	b
28	a
29	b,c
30	a
31	b
32	c
33	b,c
34	a
35	a,b
36	b
37	a
38	b
39	b,f,a,c,d,e
40	a,c,d

«УТВЕРЖДАЮ»
зав. кафедрой информатики и
программной инженерии,
доцент _____ Л.А. Тягульская
«___» _____ 2020 г.

Вопросы к зачету
по дисциплине «Введение в алгоритмы»
для студентов V курса, з/о
направления подготовки «Программная инженерия»
профиль подготовки «Разработка программно-информационных
систем»,
IX семестр

1. Задачи и алгоритмы. Массовая проблема и индивидуальная задача. Неформальное определение алгоритма.
2. Свойства алгоритма. Различные подходы к формализации понятия алгоритма.
3. Машина Тьюринга. Неформальное описание машины Тьюринга. Внешний алфавит, алфавит состояний, функциональная схема, принцип работы.
4. Вычислимые по Тьюрингу функции, основная гипотеза теории алгоритмов.
5. Рекурсивные функции. Описание класса рекурсивных функций: базисные функции (нуль-функция, функция следования и функция-проектор), операторы суперпозиции, примитивной рекурсии и минимизации.
6. Примитивно-рекурсивные и частично-рекурсивные функции. Тезис Черча.
7. Нормальные алгоритмы Маркова. Алфавит, слова, простые и заключительные формулы. Подстановки и нормальные алгоритмы Маркова.
8. Нормально вычислимые функции, принцип нормализации Маркова. Совпадение классов частично рекурсивных, нормально вычислимых и вычислимых по Тьюрингу функций.
9. Алгоритмически неразрешимые проблемы. Нумерация алгоритмов, машин Тьюринга. Проблемы распознавания самоприменимости и применимости.
10. Проблема остановки. 10-я проблема Гильберта. Проблема определения общерекурсивности алгоритма. Проблема эквивалентности алгоритмов.
11. Сравнительные оценки алгоритмов. Временная и емкостная сложность.
12. Трудоемкость алгоритма, функции оценки трудоемкости алгоритма: лучший, худший и средний случай.
13. Классификация алгоритмов по виду функции трудоемкости: количественно-зависимые по трудоемкости алгоритмы, параметрически-зависимые, количественно-параметрические, порядково-зависимые.
14. Трудоемкость основных алгоритмических конструкций.

15. Элементарные операции. Трудоемкость основных алгоритмических конструкций: следования, ветвления и цикла.
16. Методики перехода к временным оценкам: пооперационный анализ, метод Гиббсона, метод прямого определения среднего времени.
17. Сложностные классы задач. Теоретический предел трудоемкости задачи.
18. Сложностные классы задач: класс P (задачи с полиномиальной сложностью), класс NP (полиномиально проверяемые задачи), класс NPC (NP – полные задачи).
19. Построение эффективных алгоритмов. Метод декомпозиции.

Экзаменатор, преподаватель _____ Н.В. Нагаевская