

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ



И.о. декана факультета информационных
технологий и компьютерной безопасности

/ А.В. Бредихин /

19.03.2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Теория алгоритмов»

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Информационные системы и технологии цифровизации

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2022

Автор программы

С.Ю. Белецкая

И.о. заведующего кафедрой
систем

автоматизированного
проектирования и
информационных систем

П.Ю. Гусев

Руководитель ОПОП

Д.В. Иванов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Формирование у студентов системных знаний о современной теории алгоритмов, изучение формальных моделей алгоритмов, освоение методов разработки алгоритмов, получение практических навыков построения и анализа сложности алгоритмов при решении прикладных задач в информационных системах.

1.2. Задачи освоения дисциплины

- ознакомление с основными подходами к формализации понятия алгоритма;
- изучение формальных моделей алгоритмов и технологии их построения;
- получение знаний об основных алгоритмических конструкциях и способах их реализации;
- изучение методов разработки алгоритмов и типовых задач алгоритмизации;
- освоение способов описания и представления алгоритмов;
- приобретение навыков оценки сложности алгоритмов;
- получение практических навыков разработки и анализа алгоритмов при решении прикладных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Теория алгоритмов» относится к дисциплинам части, формируемой участниками образовательных отношений (дисциплина по выбору) блока Б1 учебного плана.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Теория алгоритмов» направлен на формирование следующих компетенций:

ПК-2 - Способен выполнять проектирование информационных систем и ресурсов для различных прикладных областей

ПК-6 - Способен проводить анализ качества кода и тестирование в процессе разработки информационных систем

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-2	Знать формальные модели алгоритмов, методы построения и анализа алгоритмов
	Уметь разрабатывать алгоритмы для решения прикладных задач и проводить оценку их эффективности
	Владеть навыками алгоритмизации задач проектирования информационных систем

ПК-6	Знать принципы структурного подхода к алгоритмизации, этапы проектирования алгоритмов
	Уметь производить тестирование и верификацию алгоритмов при решении прикладных задач
	Владеть навыками программной реализации алгоритмов и анализа качества кода в процессе разработки информационных систем

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Теория алгоритмов» составляет 3 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Вид учебной работы	Всего часов	Семестры			
		5			
Аудиторные занятия (всего)	72	72			
В том числе:					
Лекции	36	36			
Практические занятия (ПЗ), в том числе в форме практической подготовки (<i>при наличии</i>)					
Лабораторные работы (ЛР), в том числе в форме практической подготовки (<i>при наличии</i>)	36	36			
Самостоятельная работа	36	36			
Курсовой проект (работа) (есть, нет)	нет	нет			
Контрольная работа (есть, нет)	нет	нет			
Вид промежуточной аттестации (зачет, зачет с оценкой, экзамен)	зачёт	зачёт			
Общая трудоемкость час	108	108			
	зач. ед.	3	3		

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Ла б. зан .	СРС	Всего, час
1	Понятие алгоритма. Свойства	Интуитивное понятие алгоритма. Классификация	4	4	4	12

	алгоритмов	сификация алгоритмов. Свойства алгоритмов. Требования к алгоритмам в информационных системах. Понятие вычислимой функции, разрешимого множества. Подходы к формализации понятия «алгоритм».				
2	Формальные модели алгоритмов	Теория рекурсивных функций. Простейшие функции. Операции суперпозиции, примитивной рекурсии, минимизации. Примитивно-рекурсивные функции и их свойства. Примеры примитивно-рекурсивных функций. Частично рекурсивные функции. Общерекурсивные функции. Тезис Черча. Структура рекурсивных алгоритмов. Формализация понятия алгоритма на примере машин Тьюринга. Понятие машины Тьюринга. Команды машины Тьюринга. Построение программ для машины Тьюринга. Примеры программ. Алгоритмическая машина Поста. Принципы построения машины Поста. Команды машины Поста. Построение программ для машины Поста. Нормальные алгоритмы Маркова. Формализация понятия алгоритма на примере нормальных алгоритмов Маркова. Алфавит, буква, слово. Смежные слова. Понятие нормального алгоритма.	10	10	10	30

		Нормализуемый алгоритм. Способы композиции нормальных алгоритмов. Примеры нормальных алгоритмов.				
3	Методы построения алгоритмов	Структурный подход к проектированию алгоритмов. Основные алгоритмические конструкции. Базовые типы данных. Линейные вычислительные алгоритмы. Ветвление и циклы в вычислительных алгоритмах. Вспомогательные алгоритмы. Формы представления алгоритма: словесная, графическая, псевдокод, программа. Построение блок-схем алгоритмов. Разработка спецификаций. Этапы проектирования и анализа алгоритмов. Восходящее и нисходящее проектирование. Декомпозиция. Структурные методы алгоритмизации. Тестирование и верификация. Поиск ошибок в алгоритме. Основные типы задач при построении алгоритмов: сортировка, поиск, обработка строк, задачи теории графов, комбинаторные задачи, геометрические задачи, численные задачи. Решение типовых задач.	10	10	10	30
4	Анализ сложности алгоритмов	Понятие сложности алгоритма. Временная и объемная сложность алгоритмов. Трудоемкость алгоритма, функции оценки трудоемко-	8	8	8	24

		сти алгоритма: лучший, худший и средний случай. Классификация алгоритмов по виду функции трудоемкости. Трудоемкость основных алгоритмических конструкций: следования, ветвления и цикла. Оценка ёмкостной и временной сложности алгоритма. Сложностные классы задач: классы P, NP, NPC. Примеры NP-полных задач.				
5	Алгоритмически неразрешимые проблемы	Алгоритмические проблемы. Существование алгоритмически невычислимой функции. Примеры алгоритмически неразрешимых проблем. Проблема распознавания, самоприменимости, проблема останова.	4	4	4	12
Итого			36	36	36	108

5.2 Перечень лабораторных работ

- 1 Построение и анализ рекурсивных алгоритмов.
- 2 Разработка алгоритмов поиска и сортировки данных.
- 3 Программная реализация алгоритмов на графах.
- 4 Анализ сложности алгоритмов различной структуры.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-2	Знать формальные модели алгоритмов, методы построения и анализа алгоритмов	Знание основных подходов к формализации понятия алгоритма. Активная работа в ходе лабораторного практикума. Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь разрабатывать алгоритмы для решения прикладных задач и проводить оценку их эффективности	Знание основных алгоритмических конструкций, типовых задач алгоритмизации, методов построения и анализа алгоритмов. Решение стандартных практических задач. Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками алгоритмизации задач проектирования информационных систем	Решение прикладных задач в конкретной предметной области, выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-6	Знать принципы структурного подхода к	Знание технологии разработки алго-	Выполнение работ в срок, предусмотр-	Невыполнение работ в срок, предусмотренный в ра-

	алгоритмизации, этапы проектирования алгоритмов	ритмов на основе структурного подхода. Ответы на теоретические вопросы при защите лабораторных работ	ренный в рабочих программах	бочих программах
	Уметь производить тестирование и верификацию алгоритмов при решении прикладных задач	Умение производить тестирование и поиск ошибок в алгоритмах. Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками программной реализации алгоритмов и анализа качества кода в процессе разработки информационных систем	Решение прикладных задач при разработке компонентов информационных систем с использованием разработанного программного обеспечения, выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-2	Знать формальные модели алгоритмов, методы построения и анализа алгоритмов	Тест	Выполнение теста на 70-100%	Выполнение менее 70%

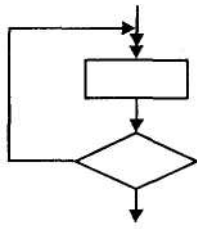
	Уметь разрабатывать алгоритмы для решения прикладных задач и проводить оценку их эффективности	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками алгоритмизации задач проектирования информационных систем	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
ПК-6	Знать принципы структурного подхода к алгоритмизации, этапы проектирования алгоритмов	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь производить тестирование и верификацию алгоритмов при решении прикладных задач	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками программной реализации алгоритмов и анализа качества кода в процессе разработки информационных систем	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

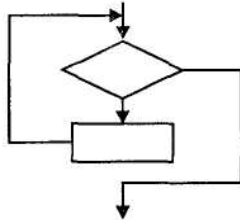
1. На рисунке представлена часть блок-схемы. Как она называется:

- 1) альтернатива;
- 2) композиция;
- 3) цикл с предусловием;
- 4) цикл с постусловием?



2. На рисунке представлена часть блок-схемы. Как она называется:

- 1) альтернатива;
- 2) композиция;
- 3) цикл с постусловием;
- 4) цикл с предусловием?



3. Сколько существует команд у машины Поста

- 1) 2;
- 2) 4;
- 3) 6;
- 4) 8

4. Рекурсия в алгоритме будет прямой, когда:

- 1) рекурсивный вызов данного алгоритма происходит из вспомогательного алгоритма, к которому в данном алгоритме имеется обращение;
- 2) порядок следования команд определяется в зависимости от результатов проверки некоторых условий;
- 3) команда обращения алгоритма к самому себе находится в самом алгоритме;
- 4) один вызов алгоритма прямо следует за другим.

5. Рекурсия в алгоритме будет косвенной, когда:

- 1) порядок следования команд определяется в зависимости от результатов проверки некоторых условий;
- 2) рекурсивный вызов данного алгоритма происходит из вспомогательного алгоритма, к которому в данном алгоритме имеется обращение;
- 3) команда обращения алгоритма к самому себе находится в самом алгоритме;
- 4) один вызов алгоритма прямо следует за другим.

6. В алгоритме Маркова ассоциативным исчислением называется:

- 1) совокупность всех слов в данном алфавите;
- 2) совокупность всех допустимых систем подстановок;
- 3) совокупность всех слов в данном алфавите вместе с допустимой системой подстановок;
- 4) когда все слова в алфавите являются смежными.

7. В ассоциативном счислении два слова называются смежными:

- 1) если одно из них может быть преобразовано в другое применением подстановок;

- 2) если одно из них может быть преобразовано в другое однократным применением допустимой подстановки;
- 3) когда существует цепочка от одного слова к другому и обратно;
- 4) когда они дедуктивны.

8 Свойство алгоритма обеспечения решения не одной задачи, а целого класса задач данного типа:

- 1) понятность;
- 2) определенность;
- 3) дискретность;
- 4) массовость.
- 5) универсальность

9. Свойство алгоритма, что при точном исполнении всех предписаний процесс должен прекратиться за конечное число шагов с определенным ответом на поставленную задачу:

- 1) понятность;
- 2) детерминированность;
- 3) дискретность;
- 4) результативность.

10 Что называют служебными словами в алгоритмическом языке:

- 1) слова, употребляемые для записи команд, входящих в СКИ;
- 2) слова, смысл и способ употребления которых задан раз и навсегда;
- 3) вспомогательные алгоритмы, которые используются в составе других алгоритмов;
- 4) константы с постоянным значением

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Команда машины Поста имеет структуру pKm , где:

- 1) p — действие, выполняемое головкой; K — номер следующей команды, подлежащей выполнению; m — порядковый номер команды;
- 2) p — порядковый номер команды; K — действие, выполняемое головкой; m — номер следующей команды, подлежащей выполнению;
- 3) p — порядковый номер команды; K — номер следующей команды, подлежащей выполнению; m — действие, выполняемое головкой;
- 4) p — порядковый номер команды; K — действие, выполняемое головкой; m — номер клетки, с которой данную команду надо произвести.

2. В машине Поста останов будет результативным:

- 1) при выполнении недопустимой команды;
- 2) если машина не останавливается никогда;
- 3) если результат выполнения программы такой, какой и ожидался;
- 4) по команде «Стоп».

3. В машине Тьюринга состояниями являются:

- 1) $\{a_0 0, a_0 1, a_0 2, \dots, a_0 t\}$;
- 2) $\{q_1, q_2, q_3, \dots, q_s\}$;
- 3) $\{q_1, q_2, q_3, \dots, q_s, a_0 0, a_0 1, a_0 2, \dots, a_0 t\}$;
- 4) $\{q_0, q_1, q_2, \dots, q_s\}$.

4. В машине Тьюринга предписание L для лентопротяжного механизма означает:

- 1) переместить ленту вправо;
- 2) переместить ленту влево;
- 3) остановить машину;
- 4) занести в ячейку символ.

5. Способ композиции нормальных алгоритмов будет итерацией, если:

- 1) выходное слово первого алгоритма является входным для второго;
- 2) существует алгоритм C, преобразующий любое слово p, содержащееся в пересечении областей определения алгоритмов A и B;
- 3) алгоритм D будет суперпозицией трех алгоритмов ABC, причем область определения D является пересечением областей определения алгоритмов A B к 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;
- 4) 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.

6 В алгоритмах Маркова дана система подстановок в алфавите $\Pi = \{a, b, c\}$:

abc — c
ba — cb
ca — ab

Преобразуйте с помощью этой системы слово bасаabc:

- 1) cbc;
- 2) ccbcbbc;
- 3) cbacba;
- 4) cbabc.

7. В алгоритмах Маркова дана система подстановок в алфавите $A = \{a, b, c\}$:

cb — abc
bac — ac
cab — b

Преобразуйте с помощью этой системы слово bсаbacab:

- 1) ccb;
- 2) cab;
- 3) cbc;
- 4) bсаab.

8. Способ композиции нормальных алгоритмов будет суперпозицией, если:

- 1) выходное слово первого алгоритма является входным для второго;
- 2) существует алгоритм C , преобразующий любое слово p , содержащееся в пересечении областей определения алгоритмов A и B ;
- 3) 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 $C(p) \neq e$, где e — пустая строка;
- 4) 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 .

9 В машине Тьюринга предписание R для лентопротяжного механизма означает:

- 1) переместить ленту вправо;
- 2) переместить ленту влево;
- 3) остановить машину;
- 4) занести в ячейку символ.

10 В алгоритме Маркова дана цепочка $P, P_1, P_2, \dots, P_n$. Если слова $P_1, P_2, \dots, P_n$ смежные, то цепочка называется:

- 1) Ассоциативной
- 2) Эквивалентной
- 3) Индуктивной
- 4) Дедуктивной

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Дан фрагмент программного кода. Оценить сложность алгоритма

```
for i:=1 to N do
begin
max:=A[i,1];
for j:=1 to N do
begin
if A[i,j]>max then
max:=A[i,j]
end;
writeln(max);
end;
```

- 1) $O(N^2)$
- 2) $O(N)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(1)$

2. Оценить сложность рекурсивного алгоритма

```
function Factorial(n: Word): integer;  
begin  
  if n > 1 then  
    Factorial:=n*Factorial(n-1)  
  else  
    Factorial:=1;  
  end;
```

- 1) $O(N)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(1)$

Ответ $O(N)$

3. Оценить сложность алгоритма многократной рекурсии

```
procedure DoubleRecursive(N: integer);  
begin  
  if N>0 then  
    begin  
      DoubleRecursive(N-1);  
      DoubleRecursive(N-1);  
    end;  
  end;
```

- 1) $O(2^N)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(N)$
- 5) $O(N^3)$

4. Дан фрагмент программного кода. Оценить сложность алгоритма

```
function Locate(data: integer): integer;  
var  
  i: integer;  
  fl: boolean;  
begin  
  fl:=false; i:=1;  
  while (not fl) and (i<=N) do  
    begin  
      if A[i]=data then  
        fl:=true  
      else  
        i:=i+1;  
      end;  
    end;  
  if not fl then
```

```
i:=0;  
Locate:=I;  
end;
```

- 1) $O(N)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(N^3)$

5. Дан фрагмент программного кода. Определить вычислительную сложность алгоритма

```
procedure Slow;  
var  
i,j,k: integer;  
begin  
for i:=1 to N do  
for j:=1 to N do  
for k:=1 to N do  
{действие}  
end;  
procedure Fast;  
var  
i,j: integer;  
begin  
for i:=1 to N do  
for j:=1 to N do  
Slow;  
end;  
procedure Both;  
begin  
Fast;  
end;
```

- 1) $O(N^5)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(N^3)$

6. Составить алгоритм перемножения двух матриц порядка N и оценить его сложность

- 1) $O(N^3)$
- 2) $O(N^5)$

- 3) $O(N^2)$
- 4) $O(\log N)$
- 5) $O(2^N)$

7. Построить алгоритм пузырьковой сортировки и оценить его сложность

- 1) $O(N^2)$
- 2) $O(N^5)$
- 3) $O(N^3)$
- 4) $O(\log N)$
- 5) $O(2^N)$

8. Оценить сложность алгоритма поиска наименьшего элемента в неотсортированном массиве

- 1) $O(N)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(N^3)$

9. Построить алгоритм определения чётности целого числа, представленного в двоичном виде и оценить его сложность

- 1) $O(1)$
- 2) $O(N^2)$
- 3) $O(\log N)$
- 4) $O(2^N)$
- 5) $O(N)$

10. Дан фрагмент программного кода. Оценить сложность алгоритма.

```
procedure Slow;
var
  i, j, k: integer;
begin
  for i:=1 to N do
    for j:=1 to N do
      for k:=1 to N do
        {действие}
      end;
    end;
  end;
procedure Fast;
```

```

var
i,j: integer;
begin
for i:=1 to N do
for j:=1 to N do
{ действие}
end;
procedure Both;
begin
Fast;
Slow;
end;

```

- 1) $O(N^3)$
- 2) $O(N^5)$
- 3) $O(N^2)$
- 4) $O(\log N)$
- 5) $O(2^N)$

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Интуитивное понятие алгоритма. Классификация алгоритмов.
2. Свойства алгоритмов. Требования к алгоритмам в информационных системах.
3. Понятие вычислимой функции, разрешимого множества. Подходы к формализации понятия «алгоритм».
4. Теория рекурсивных функций. Простейшие функции. Операции суперпозиции, примитивной рекурсии, минимизации.
5. Примитивно-рекурсивные функции и их свойства. Примеры примитивно-рекурсивных функций.
6. Частично рекурсивные функции. Общерекурсивные функции. Тезис Черча.
7. Структура рекурсивных алгоритмов.
8. Понятие машины Тьюринга. Команды машины Тьюринга. Построение программ для машины Тьюринга.
9. Алгоритмическая машина Поста. Принципы построения машины Поста. Команды машины Поста. Построение программ для машины Поста.
10. Нормальные алгоритмы Маркова. Построение нормальных алгоритмов. Способы композиции нормальных алгоритмов.
11. Структурный подход к проектированию алгоритмов. Основные алгоритмические конструкции.
12. Базовые типы данных при построении алгоритмов.
13. Линейные вычислительные алгоритмы. Ветвление и циклы в вычислительных алгоритмах. Вспомогательные алгоритмы.
14. Формы представления алгоритма. Построение блок-схем алгоритмов.
15. Этапы проектирования и анализа алгоритмов. Восходящее и нисходящее проектирование. Декомпозиция.
16. Структурные методы алгоритмизации. Тестирование и верификация. Поиск ошибок в алгоритме.
17. Решение задач сортировки при построении алгоритмов.
18. Решение задач поиска при построении алгоритмов.
19. Основные задачи теории графов.
20. Понятие сложности алгоритма. Временная и объемная сложность алгоритмов. Оценка сложности алгоритма.

21. Классификация алгоритмов по виду функции трудоемкости. Трудоемкость основных алгоритмических конструкций: следования, ветвления и цикла.
22. Сложностные классы задач: классы P, NP, NPC.
23. NP-полные задачи.
24. Алгоритмически неразрешимые проблемы.

7.2.5 Примерный перечень вопросов для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачёт проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Зачёт ставится в случае, если студент набрал более 10 баллов.
2. Незачёт ставится в случае, если студент набрал менее 10 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Понятие алгоритма. Свойства алгоритмов	ПК-2, ПК-6	Тест, защита лабораторных работ
2	Формальные модели алгоритмов	ПК-2, ПК-6	Тест, защита лабораторных работ
3	Методы построения алгоритмов	ПК-2, ПК-6	Тест, защита лабораторных работ
4	Анализ сложности алгоритмов	ПК-2, ПК-6	Тест, защита лабораторных работ
5	Алгоритмически неразрешимые проблемы	ПК-2, ПК-6	Тест, защита лабораторных работ

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Алябьева В.Г. Теория алгоритмов [Электронный ресурс]: учеб. пособие. – Пермь: ПГПУ, 2013. – 125 с. Режим доступа:
<http://www.iprbookshop.ru/32100>
2. Собенина О.В. Теория алгоритмов [Электронный ресурс]: Учеб. пособие. Электрон. текстовые, граф. дан. – Воронеж: ВГТУ, 2011 (1 файл)
3. Судоплатов С.В. Математическая логика и теория алгоритмов: учебник. – Новосибирск: НГТУ, 2008. – 224 с.
4. Холопкина Л.В. Математическая логика и теория алгоритмов: учеб. пособие. – Воронеж: ВГТУ, 2008. – 162 с.
5. Дехтярь М.И. Лекции по дискретной математике [Электронный ресурс]: учеб. Пособие. – Москва: ИНТУИТ, 2016. – 181 с. Режим доступа:
<http://www.iprbookshop.ru/62815>
6. Хагарт Р. Дискретная математика для программистов [Электронный ресурс]: учеб. пособие. – М.: Техносфера, 2012. – 400 с. – Режим доступа:
<http://www.iprbookshop.ru/12723.html>

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Программное обеспечение

Microsoft Visual Studio Community

Ресурсы информационно-телекоммуникационной сети «Интернет»

<http://www.edu.ru/>

Образовательный портал ВГТУ

Информационные справочные системы

<http://window.edu.ru>

<https://wiki.cchgeu.ru/>

Современные профессиональные базы данных

ХабрХабр <https://habr.com/ru/>

Исходники.ru <https://sources.ru/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория , оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой
--

Дисплейный класс , оснащенный компьютерными программами для проведения лабораторного практикума
--

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Теория алгоритмов» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометить важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим раз-

	делом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.

11 Лист регистрации изменений

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
1	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины	31.08.2020	
2	Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем	31.08.2020	
3	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины	31.08.2021	
4	Внесены изменения в связи с вступлением в силу приказа № 403-ФЗ от 2.12.2019 «О внесении изменений в Федеральный закон об образовании в Российской Федерации» и отдельные законодательные акты Российской Федерации	31.08.2021	

