

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет промышленных технологий и дизайна»
(СПбГУПТД)

УТВЕРЖДАЮ
Первый проректор, проректор
по УР

_____ А.Е. Рудин

Рабочая программа дисциплины

Б1.О.20

Теория алгоритмов

Учебный план: 2026-2027 09.03.04 ИИТА Программ инженерии ОО №1-1-195.plx

Кафедра:

33

Цифровых и аддитивных технологий

Направление подготовки:
(специальность)

09.03.04 Программная инженерия

Профиль подготовки:
(специализация)

Программная инженерия

Уровень образования:

бакалавриат

Форма обучения:

очная

План учебного процесса

Семестр (курс для ЗАО)		Контактная работа обучающихся		Сам. работа	Контроль, час.	Трудоёмкость, ЗЕТ	Форма промежуточной аттестации
		Лекции	Практ. занятия				
5	УП	32	32	43,75	0,25	3	Зачет
	РПД	32	32	43,75	0,25	3	
Итого	УП	32	32	43,75	0,25	3	
	РПД	32	32	43,75	0,25	3	

Рабочая программа дисциплины составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.04 Программная инженерия, утверждённым приказом Минобрнауки России от 19.09.2017 г. № 920

Составитель (и):

к.т.н., Заведующий кафедрой

Сошников Антон
Владимирович

От кафедры составителя:

Заведующий кафедрой цифровых и аддитивных технологий

Сошников Антон
Владимирович

От выпускающей кафедры:

Заведующий кафедрой

Сошников Антон
Владимирович

Методический отдел:

1 ВВЕДЕНИЕ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

1.1 Цель дисциплины: Формирование у обучающихся теоретико-методологической базы для анализа, проектирования и оценки алгоритмических решений, а также развитие навыков строгого формального рассуждения о вычислимости, сложности и эффективности алгоритмов в контексте реальных технических и ресурсных ограничений.

1.2 Задачи дисциплины:

Сформировать представление о фундаментальных моделях вычислений (машины Тьюринга, рекурсивные функции) и их роли в определении понятия алгоритма.

Овладеть методами анализа временной и пространственной сложности алгоритмов и классификацией задач по сложности.

Развить способность формально оценивать разрешимость и вычислимость задач, а также обосновывать невозможность эффективного решения отдельных проблем.

Обеспечить освоение классических алгоритмических парадигм («разделяй и властвуй», динамическое программирование, жадные алгоритмы, перебор с возвратом).

Сформировать навыки сравнительного тестирования и экспериментальной оценки эффективности алгоритмов в реальных вычислительных средах.

1.3 Требования к предварительной подготовке обучающегося:

Предварительная подготовка предполагает создание основы для формирования компетенций, указанных в п. 2, при изучении дисциплин:

Алгоритмизация и программирование

Теория систем и системный анализ

Математика

Системное программирование

Прикладная статистика

2 КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

УК-2: Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений					
Знать: Основные виды вычислительных ресурсов и типы ограничений, влияющих на выбор и эффективность алгоритмических решений.					
Уметь: Проводить формальный анализ поставленной задачи с целью выявления требуемых алгоритмических решений и определения круга подзадач.					
Владеть: Навыками оценки ресурсных требований алгоритмов и анализа влияния технических и внешних ограничений на их реализацию.					
ОПК-1: Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности					
Знать: Базовые модели вычислений, формальные языки и классы алгоритмической сложности.					
Уметь: Использовать методы математического анализа, логического рассуждения и алгоритмического моделирования при проектировании и оценке алгоритмов.					
Владеть: Навыками теоретического анализа алгоритмических моделей и вычислительных процессов.					
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов					
Знать: Фундаментальные классы алгоритмов, методы их анализа, модели вычислений.					
Уметь: Применять алгоритмические методы при разработке компонентов информационных систем и обработке структурированных данных в базах и хранилищах.					
Владеть: Навыками реализации, отладки и сравнительного тестирования алгоритмических прототипов, включая оценку корректности и эффективности их работы в различных вычислительных средах.					
ОПК-7: Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой					
Знать: Формальные модели вычислений (машины Тьюринга, рекурсивные функции), классы сложности алгоритмов и критерии разрешимости вычислительных задач.					
Уметь: Проводить анализ вычислительной сложности алгоритмов и обосновывать невозможность или неэффективность решения отдельных задач в рамках заданных ограничений.					
Владеть: Навыками формального рассуждения о свойствах алгоритмов, позволяющим строить строгие доказательства корректности и границ применимости вычислительных методов.					

3 РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Наименование и содержание разделов, тем и учебных занятий	Семестр (курс для ЗАО)	Контактная работа		СР (часы)	Инновац. формы занятий	Форма текущего контроля
		Лек. (часы)	Пр. (часы)			

Раздел 1. Основы вычислимости и теории сложности	5					О
Тема 1. Понятие алгоритма. История и формализация. Машина Тьюринга как универсальная модель вычислений. Практические занятия: Построение и анализ простых машин Тьюринга для решения базовых задач (сложение, распознавание языков).		4	4	5		
Тема 2. Частично рекурсивные функции. Тезис Чёрча–Тьюринга. Примеры невычислимых функций. Практические занятия: Реализация рекурсивных и частично рекурсивных функций; демонстрация границ вычислимости.		4	4	5		
Тема 3. Проблема останова. Теорема о неразрешимости. Другие неразрешимые задачи (эквивалентность программ, тотальность). Практические занятия: Доказательство неразрешимости задачи останова; построение редукций к другим неразрешимым проблемам.		4	4	5		
Тема 4. Основы теории сложности: временная и пространственная сложность. Классы P, NP, co-NP, EXP. Полиномиальная сводимость. Практические занятия: Анализ временной сложности алгоритмов (итеративных и		4	4	5		О
Раздел 2. Алгоритмические парадигмы и прикладной анализ						
Тема 5. Классические алгоритмические стратегии: «разделяй и властвуй», жадные алгоритмы, динамическое программирование. Практические занятия: Реализация и сравнение алгоритмов сортировки и поиска по стратегиям «разделяй и властвуй» и жадности.		4	4	5		
Тема 6. Перебор с возвратом (backtracking) и методы оптимизации поиска. Задачи комбинаторной оптимизации. Практические занятия: Решение задач (например, задача о рюкзаке, раскраска графа) методами динамического программирования и перебора с возвратом.		4	4	5		
Тема 7. NP-полнота. Примеры NP-полных задач (SAT, CLIQUE, HAMPATH, TSP). Стратегии работы с NP-трудными задачами. Практические занятия: Построение полиномиальных сводимостей между NP-полными задачами; анализ практической сложности на реальных данных.		4	4	5		
Тема 8. Эмпирическая оценка алгоритмов: профилирование, сравнительное тестирование, влияние кэширования, локальности данных и аппаратной платформы. Практические занятия: Профилирование и сравнительное тестирование двух реализаций одной задачи		4	4	8,75		

Итого в семестре (на курсе для ЗАО)		32	32	43,75		
Консультации и промежуточная аттестация (Зачет)		0,25				
Всего контактная работа и СР по дисциплине		64,25		43,75		

4 КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Курсовое проектирование учебным планом не предусмотрено

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1 Описание показателей, критериев и системы оценивания результатов обучения

5.1.1 Показатели оценивания

Код компетенции	Показатели оценивания результатов обучения	Наименование оценочного средства
УК-2	Раскрывает сущность основных вычислительных ресурсов. Формирует задачу и выявить ресурсные ограничения. Обосновывает выбранный алгоритм с учётом правовых, технических и временных рамок.	
ОПК-1	Классифицирует базовые модели вычислений, формальные языки и классы алгоритмической сложности. Использует математический аппарат для анализа алгоритмов. Демонстрирует логическое рассуждение при доказательстве свойств вычислений.	
ОПК-6	Объясняет особенности фундаментальных классов алгоритмов. Реализует и тестирует алгоритмы на корректность и эффективность. Демонстрирует результат сравнения реализации профессиональной задачи в различных вычислительных средах.	
ОПК-7	Раскрывает сущность формальных моделей вычислений. Работает с формальными моделями вычислений (машина Тьюринга, рекурсия). Демонстрирует заключение о неразрешимости и сложности задачи строго и корректно.	

5.1.2 Система и критерии оценивания

Шкала оценивания	Критерии оценивания сформированности компетенций	
	Устное собеседование	Письменная работа
Зачтено	Обучающийся своевременно выполнил практические задания и представил результаты в форме презентации; отвечает на теоретический вопрос по материалам лекций, возможно допуская несущественные ошибки. Учитываются баллы, накопленные в течение семестра.	
Не зачтено	Обучающийся своевременно не выполнил (выполнил частично) практические задания и не представил результаты в форме презентации; при ответе на вопрос преподавателя допустил существенные ошибки. Не учитываются баллы, накопленные в течение семестра.	

5.2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности

5.2.1 Перечень контрольных вопросов

№ п/п	Формулировки вопросов
Семестр 5	
1	Что такое алгоритм? Какие свойства он должен обладать?

2	В чём состоит тезис Чёрча–Тьюринга? Почему он важен?
3	Опишите устройство и работу одноленточной машины Тьюринга.
4	Постройте машину Тьюринга, распознающую язык $\{0^n 1^n \mid n \geq 0\}$.
5	Что такое частично рекурсивная функция? Чем она отличается от общерекурсивной?
6	Приведите пример функции, не являющейся вычислимой.
7	Сформулируйте проблему останова. Почему она неразрешима?
8	Что такое редукция задач? Как она используется для доказательства неразрешимости?
9	Докажите, что проблема эквивалентности программ неразрешима.
10	Что такое временная сложность алгоритма? Как определяются классы O, Ω , Θ ?
11	Почему важно различать худший, средний и лучший случаи сложности?
12	Как определяется класс P? Приведите примеры задач из P.
13	Что такое класс NP? Как проверяется принадлежность задачи к NP?
14	В чём разница между NP и co-NP?
15	Что означает полиномиальная сводимость задачи A к задаче B?
16	Почему класс EXP строго шире P?
17	Может ли NP-полная задача быть решена за полиномиальное время? Что это означало бы?
18	Как связаны понятия разрешимости и сложности?
19	Что такое универсальная машина Тьюринга?
20	Почему существуют задачи, которые алгоритмически неразрешимы?
21	В чём суть парадигмы «разделяй и властвуй»? Приведите пример (сортировка слиянием, быстрая сортировка).
22	Как работает жадный алгоритм? Всегда ли он даёт оптимальное решение?
23	Опишите принцип динамического программирования. Чем он отличается от рекурсии?
24	Какие свойства задачи позволяют применить динамическое программирование?
25	Приведите пример NP-полной задачи и объясните, почему она NP-полна. Как доказать NP-полноту новой задачи?
26	Что означает, что задача NP-трудная? Чем она отличается от NP-полной? Какие подходы используются для решения NP-трудных задач на практике?
27	Что такое эвристики и аппроксимационные алгоритмы?
28	Как оценивается эффективность алгоритма экспериментально?
29	Какие метрики производительности вы используете при профилировании?
30	Почему два алгоритма с одинаковой асимптотикой могут сильно отличаться по времени выполнения?
31	Как влияет локальность данных на производительность алгоритма?
32	Как кэширование процессора влияет на выбор структур данных (массив vs список)?
33	Можно ли считать алгоритм «эффективным», если он работает быстро на конкретных данных, но имеет экспоненциальную сложность в худшем случае?
34	Как выбрать алгоритм при наличии ограничений по памяти и времени?
35	Как проводится сравнительное тестирование двух реализаций одной задачи?
36	Какие инструменты профилирования вы применяли (perf, gprof, time, cProfile)?
37	Как технические ограничения (разрядность, ОЗУ, энергопотребление) влияют на выбор алгоритма в реальных системах?

5.2.2 Типовые тестовые задания

Не предусмотрено.

5.2.3 Типовые практико-ориентированные задания (задачи, кейсы)

1. Реализация машины Тьюринга в симуляторе или на Python для распознавания простого регулярного языка (например, чётное число нулей).
2. Формальное доказательство неразрешимости задачи: «существует ли вход, на котором программа закичивается?» через сведение к проблеме останова.
3. Реализация и сравнение трёх алгоритмов сортировки (быстрая, слиянием, пирамидальная) по времени и потреблению памяти на разных наборах данных.
4. Решение задачи о рюкзаке двумя способами: жадным алгоритмом и динамическим программированием — сравнить точность и время.

5.3 Методические материалы, определяющие процедуры оценивания знаний, умений, владений (навыков и (или) практического опыта деятельности)

5.3.1 Условия допуска обучающегося к промежуточной аттестации и порядок ликвидации академической задолженности

Проведение промежуточной аттестации регламентировано локальным нормативным актом СПбГУПТД «Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся»

5.3.2 Форма проведения промежуточной аттестации по дисциплине

Устная ☐ Письменная ☐ Компьютерное тестирование ☐ Иная ☐

5.3.3 Особенности проведения промежуточной аттестации по дисциплине

Время на подготовку к устному собеседованию составляет 30 минут;
Выполнение тестовых заданий осуществляется за 30 минут.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1 Учебная литература

Автор	Заглавие	Издательство	Год издания	Ссылка
6.1.1 Основная учебная литература				
Когабаев, Н. Т.	Дискретная математика и теория алгоритмов	Новосибирск: Новосибирский государственный университет	2023	https://www.iprbooks.hop.ru/134568.html
Богатилов, В. Н., Павлов, В. А.	Элементы теории алгоритмов и формальных языков	Тверь: Тверской государственный технический университет	2023	https://www.iprbooks.hop.ru/151314.html
6.1.2 Дополнительная учебная литература				
Мачикина, Е. П.	Математическая логика и теория алгоритмов	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики	2020	https://www.iprbooks.hop.ru/102154.html
Мачикина, Е. П., Новожилов, Д. И.	Задачник по математической логике и теории алгоритмов	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики	2023	https://www.iprbooks.hop.ru/138822.html

6.2 Перечень профессиональных баз данных и информационно-справочных систем

Информационная система «Единое окно доступа к образовательным ресурсам. Раздел. Информатика и информационные технологии» [Электронный ресурс]. URL: http://window.edu.ru/catalog/?p_rubr=2.2.75.6
Электронно-библиотечная система IPRbooks [Электронный ресурс]. URL: <http://www.iprbookshop.ru/>

6.3 Перечень лицензионного и свободно распространяемого программного обеспечения

MicrosoftOfficeProfessional
Microsoft Windows

6.4 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Аудитория	Оснащение
Компьютерный класс	Мультимедийное оборудование, компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду
Лекционная аудитория	Мультимедийное оборудование, специализированная мебель, доска