|  |
| --- |
| Министерство науки и высшего образования Российской Федерации |
| федеральное государственное автономное образовательное учреждение  высшего образования |
| **«Национальный исследовательский ядерный университет «МИФИ»** |
| ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ |
| КАФЕДРА КИБЕРНЕТИКИ (№ 22) |

ОДОБРЕНО

протокол № 18 / 03  
от « 31 » мая 2018 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

**Дискретная математика (Теория алгоритмов и сложность вычислений)**

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.03.04 Программная инженерия |
|  |  |
| Профиль подготовки (при его наличии) |  |
|  |  |
| Наименование образовательной программы (специализация) | Математическое и программное обеспечение вычислительных машин и компьютерных сетей |
|  |  |
| Квалификация (степень) выпускника | бакалавр |
|  |  |
| Форма обучения | очная |

г. Москва, 2018 г.

**ПАСПОРТ**

**фонда оценочных средств**

**по дисциплине**

**«Дискретная математика (Теория алгоритмов и сложность вычислений)»**

(наименование дисциплины)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 3 |  | 4 | 144 | 32 | 32 | 0 | 44 | 0 | Э |
| ИТОГО | 0 | 4 | 144 | 32 | 32 | 0 | 44 | 0 |  |

Группа: Б18-504, Б18-514

**Модели контролируемых компетенций**

В результате освоения дисциплины у выпускника формируются следующие компетенции:

ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

УК-1 – Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

**Программа оценивания контролируемых компетенций**

Формирование у студентов компетенций контролируется в течение всего времени освоения дисциплины в рамках:

* текущего контроля;
* рубежного контроля;
* промежуточного контроля.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п.** | **Наименование раздела учебной дисциплины** | **Неде-ли** | **Лек-ции, час.** | **Практ. зан./ семи-нары, час.** | **Лаб. рабо-ты, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттеста-ция раздела (форма\*, неделя)** | **Макси-мальный балл за раздел \*\*** | **Компетенции по разделам, проверяемые при текущем и рубежном контроле** | **Компетенции, проверяемые на зач. /экз.** |
| **3 семестр** | | | | | | | | | | |
| 1 | Формальные описания алгоритмов | 1-5 | 10 | 10 |  | КР-3, КР-5,Т-6,Сем-5 | КИ, 6 | 22 | ОПК-1, ОПК-3, УК-1 |  |
| 2 | Числовые множества и арифметические вычисления | 6-11 | 12 | 12 |  | Сем-11, Т-11 | КИ, 12 | 12 | ОПК-1, ОПК-3, УК-1 |  |
| 3 | Рекурсивные функции | 12-16 | 10 | 10 |  | КР-14, Т-16, Сем-15 | КИ, 16 | 16 | ОПК-1, ОПК-3, УК-1 |  |
|  | Экзамен |  |  |  |  |  | Э | 50 |  | ОПК-1, ОПК-3, УК-1 |
|  | Итого за 3 семестр |  | 32 | 32 |  |  |  | 100 |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Недели** | **Содержание / Темы занятий** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** | **Компетенции по темам, проверяемые при текущем контроле** | **Виды тек.контроля по проверке компетенций** | **Компетенции по темам, проверяемые на зач. /экз.** |
| **3 семестр, лекции** | | | | | | | |
| 1-5 | **1. Формальные описания алгоритмов** | 10 | 10 | 0 |  |  |  |
| Развитие понятия алгоритма и современная теория алгоритмов. Тезис Тьюринга и классические машины Тьюринга. Модификация машин Тьюринга. Многоленточные машины. Универсальная машина. Самоанализирующие машины.  Теоремы Шеннона. Нормальные алгоритмы и тезис Маркова. Преобразование машин Тьюринга в нормальные алгоритмы.  Понятие алгоритмической разрешимости. Алгоритмически неразрешимые задачи об остановке машины Тьюринга и записи данного знака.  Эффективная перечислимость и эффективная распознаваемость множеств. Эффективное распознавание и теорема Поста. Эффективное перечисление множества машин Тьюринга и существование в нем эффективно перечислимых и эффективно нераспознаваемых подмножеств. Геделева нумерация и кодовые числа алгоритмов Маркова. | ОПК-1, ОПК-3, УК-1 | КР-3, КР-5, Сем-5,  Т-6 | ОПК-1, ОПК-3, УК-1 |
| 6-11 | **2. Числовые множества и арифметические вычисления** | 12 | 12 | 0 |  |  |  |
| Равномощные множества и кардинальные числа. Парадокс Галилея и трансфинитные числа. Конечные, счетно-бесконечные и несчетные множества.  Счетность множества натуральных, целых, рациональных и алгебраических чисел. Счетность множества пар, n-ок и комплексов натуральных чисел  Канторов диагональный процесс. Несчетность множества трансцендентных, действительных, иррациональных и комплексных чисел. Теорема Кантора. Множества еще большей мощности.  Парадоксы теории множеств. Вычислимые действительные числа. Вычислимость алгебраических чисел и существование вычислимых трансцендентных чисел. Невычислимые числа.  Арифметические функции и несчетность их множества. Вычислимые арифметические функции и невозможность их эффективного перечисления.  Частичные арифметические функции и несчетность их множества. Вычислимые частичные арифметические функции и их эффективное перечисление. Теорема Черча. Невозможность эффективного распознавания и сравнения вычислимых функций. Невычислимые функции. | ОПК-1, ОПК-3, УК-1 | Сем-11,Т-11 | ОПК-1, ОПК-3, УК-1 |
| 12-16 | **3. Рекурсивные функции и сложность вычислений** | 8 | 8 | 0 |  |  |  |
| Примитивно-рекурсивные функции и базис Клини. Частично-рекурсивные функции и расширенный базис Клини.  Общерекурсивные функции. Задание частично-рекурсивных функций при помощи системы уравнений. Сложение и мультиплицирование функций. Кусочные функции. Теорема о неявных мажорируемых функциях.  Геделева нумерация и эффективная перечислимость примитивно-рекурсивных и частично-рекурсивных функций. Невозможность эффективного перечисления общерекурсивных функций. Невозможность эффективной распознаваемости примитивно рекурсивных функций среди общерекурсивных, а также общерекурсивных функций среди частично-рекурсивных.  Непримитивно рекурсивные функции. Нерекурсивные функции. Границы применимости формальных моделей алгоритмов. | ОПК-1, ОПК-3, УК-1 | КР-14,  Сем-15, Т-16 | ОПК-1, ОПК-3, УК-1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *3 семестр*  *Семинарские занятия* | **Компетенции по темам, проверяемые при текущем контроле** | **Виды тек.контроля по проверке компетенций** | **Компетенции по темам, проверяемые на зач. /экз.** |
| 1-5 | **Формальные описания алгоритмов** | | | |
| Классические машины Тьюринга. Многоленточные машины Тьюринга. Нормальные алгоритмы. | ОПК-1, ОПК-3, УК-1 | КР-3, КР-5,Т-6 Сем-5 | ОПК-1, ОПК-3, УК-1 |
| 6-11 | **Числовые множества и арифметические вычисления** | | | |
| Бесконечные множества: счетность, перечислимость, распознаваемость.  Трансфинитные числа и парадоксы.  Числовые множества, вычислимые числа.Арифметические функции.  Частичные арифметические функции. | ОПК-1, ОПК-3, УК-1 | Сем-11, Т-11 | ОПК-1, ОПК-3, УК-1 |
| 12-16 | **Рекурсивные функции и сложность вычислений** | | | |
| Простейшие примитивно-рекурсивные функции, представление в базисе Клини.  Доказательство примитивной рекурсивности функций, восстановление функций по схеме рекурсии.Частично-рекурсивные функции. | ОПК-1, ОПК-3, УК-1 | КР-14, Сем-15, Т-16 | ОПК-1, ОПК-3, УК-1 |

**Соответствие оценочных средств видам контроля**

|  |  |
| --- | --- |
| **Вид контроля** | **Наименование оценочного средства (способ оценки: устно/ письменно /комп.технолог.)** |
| Т | Тестирование |
| КИ | Контроль по итогам |
| Сем | Семинар |
| Э | Экзамен |
| КР | Контрольная работа (письменно) |

**Правила выставления оценок за КИ**

|  |  |
| --- | --- |
| **КИ** | **Перечень контрольных мероприятий**(в скобках после наименования контрольного мероприятия указано максимальное число баллов) |
| КИ6 | КР-3 (7) + КР-6(7) + сем-5(3) + Т-6(5). Суммарно максимум 22 балла |
| КИ11 | Сем-11(3) + Т-11(9). Суммарно максимум 12 баллов |
| КИ16 | КР-14(7) + сем-15(3) + Т-16(6). Суммарно максимум 16 баллов |

**В результате освоения дисциплины студент должен:**

**Знать:**

* З-1 знать определение и основные свойства алгоритма;
* З-2 знать устройство машины Тьюринга и тезис Тьюринга;
* З-3 знать принципы построения модификаций классических машин Тьюринга;
* З-4 знать понятие алгоритмической разрешимости;
* З-5 знать понятия эффективной перечислимости и эффективной распознаваемости множеств;
* З-6 знать основы нотации нормальных алгоритмов и тезис Маркова;
* З-7 знать основные свойства бесконечных множеств и теорему Кантора;
* З-8 знать сущность и причины возникновения известных парадоксов теории множеств;
* З-9 знать определение и свойства арифметических и частичных арифметических функций;
* З-10 знать свойства вычислимых чисел и функций;
* З-11 знать суть и свойства рекурсивных функций;
* З-12 знать понятия временной и пространственной сложности алгоритмов.

**Уметь:**

* У-1 уметь строить классические и многоленточные машины Тьюринга;
* У-2 уметь строить нормальные алгоритмы Маркова;
* У-3 уметь строить доказательства эффективной перечислимости и эффективной распознаваемости изученных типов множеств;
* У-4 уметь определять счетность и эффективную перечислимость различных числовых множеств;
* У-5 уметь строить невычислимые функции;
* У-6 уметь определять счетность и эффективную перечислимость различных подклассов частичных арифметических функций;
* У-7 уметь представлять функции в стандартном и расширенном базисе Клини;
* У-8 уметь определять сложность алгоритмов.

**Владеть / быть в состоянии продемонстрировать:**

* В-1 навыки написания алгоритмов в нотации Тьюринга и Маркова;
* В-2 навыки построения схем примитивно-рекурсивных функций в базисе Клини;
* В-3 навыки восстановления функций, заданных системой рекурсивных уравнений;
* В-4 навыки логического построения доказательств теорем.

**Измерители формирования компетенций (через З, У, Н)**

**Знания**

|  |  |
| --- | --- |
| З-1 знать определение и основные свойства алгоритма; | -тест №1  -зачет |
| З-2 знать устройство машины Тьюринга и тезис Тьюринга; | -тест №1  -кр№1  -зачет |
| З-3 знать принципы построения модификаций классических машин Тьюринга; | -тест №1  -зачет |
| З-4 знать понятие алгоритмической разрешимости; | -тест №1  --зачет |
| З-5 знать понятия эффективной перечислимости и эффективной распознаваемости множеств; | -тест №1  -зачет |
| З-6 знать основы нотации нормальных алгоритмов и тезис Маркова; | -тест №1  -кр№2  -зачет |
| З-7 знать основные свойства бесконечных множеств и теорему Кантора; | -тест №2  -зачет |
| З-8 знать сущность и причины возникновения известных парадоксов теории множеств; | -тест №2  -зачет |
| З-10 знать свойства вычислимых чисел и функций; | -тест №2  -зачет |
| З-11 знать суть и свойства рекурсивных функций; | -тест №3  -кр№3  -зачет |
| З-12 знать понятия временной и пространственной сложности алгоритмов. | -зачет |

**Умения**

|  |  |
| --- | --- |
| У-1 уметь строить классические и многоленточные машины Тьюринга; | -кр№1 |
| У-2 уметь строить нормальные алгоритмы Маркова; | -кр№2 |
| У-3 уметь строить доказательства эффективной перечислимости и эффективной распознаваемости изученных типов множеств; | -тест №1  -зачет |
| У-4 уметь определять счетность и эффективную перечислимость различных числовых множеств; | -тест №1  -зачет |
| У-5 уметь строить невычислимые функции; | -тест №2  -зачет |
| У-6 уметь определять счетность и эффективную перечислимость различных подклассов частичных арифметических функций; | -тест №2  -зачет |
| У-7 уметь представлять функции в стандартном и расширенном базисе Клини; | -тест №3  -зачет |
| У-8 уметь определять сложность алгоритмов. | -зачет |

**Навыки**

|  |  |
| --- | --- |
| В-1 навыки написания алгоритмов в нотации Тьюринга и Маркова; | -кр№1  -кр№2 |
| В-2 навыки построения схем примитивно-рекурсивных функций в базисе Клини; | -кр№3 |
| В-3 навыки восстановления функций, заданных системой рекурсивных уравнений; | -тест№3  -кр№3  -зачет |
| В-4 навыки логического построения доказательств теорем. | -зачет |

**Индикаторы формирования компетенций**

|  |  |  |  |
| --- | --- | --- | --- |
| **Компетенции** | **Знания (знать)** | **Умения (уметь)** | **Навыки (владеть)** |
| УК-1 | З-4, З-5, З-7, З-10, З-11 | У-3,У-4,У-5 | В-4 |
| ОПК-1 | З-1, З-4, З-11, 3-12 | У-1, У-2, У-7, У-8 | В-1, В-2, В-3, В-4 |
| ОПК-3 | З-1, З-2, З-3, З-4, З-6, З-11 | У-1, У-2, У-7, У-8 | В-1, В-2 |

**Комплект заданий для контрольных работ по дисциплине**

**Дискретная математика (Теория алгоритмов и сложность вычислений)**

***Контрольная работа №1 (КР3)***

*Необходимо решить одну из двух задач №1 (типа «а» или «б») и одну из двух задач №2 (типа «а» или «б»). Решение следует прокомментировать.*

**Вариант 1.**

**Задача №1-а.**

Алфавит {1,2}. Все встреченные в слове комбинации «22» заменить на комбинации «11».

**Задача №1-б.**

Алфавит {1,2}. Определить, есть ли в слове два подряд вхождения символа «2». Если да, то каждую пару символов «2» заменить на пару символов «1». Если нет, то стереть все слово и записать вместо него символ «\*».

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово после разделителя «\*» в прямом порядке. При копировании удвоить все нули. Разделитель необходимо поставить самостоятельно.

**Задача №2-б.**

Алфавит {0,1}. Скопировать «скелет» слова после разделителя «\*» в прямом порядке. Разделитель необходимо поставить самостоятельно. Под «скелетом» понимается копия слова, в которой убраны все повторяющиеся символы. Например, вместо 000011100110 должно получиться 000011100110\*01010.

**Вариант 2.**

**Задача №1-а.**

Алфавит {1}. Дано число n, записанноев унарном коде. Число обязательно задано (на ленте есть как минимум одна палочка). Определить какое это число: если оно четное положительное, то поставить в конце слова «\*\*», если нечетное положительное – то поставить в конце слова «\*», если это число ноль – ничего не делать.

**Задача №1-б.**

Алфавит {1}. Дано число n, записанноев унарном коде. Число обязательно задано. Определить какое это число: если оно четное и больше трех, то поставить в конце слова «\*\*», если нечетное и больше трех – то поставить в конце слова «\*», если это число меньше или равно трем – стереть все слово и поставить вместо него «\*\*\*».

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово в обратном порядке без всякого разделителя.При копировании удвоить все «0».

**Задача №2-б.**

Алфавит {0,1}. Скопировать слово в обратном порядке без всякого разделителя.При копировании удвоить каждый второй встреченный ноль (символы отсчитываются начиная с конца слова).

**Вариант 3.**

**Задача №1-а.**

Алфавит {a,b}. Все встреченные в слове комбинации «ab» заменить на комбинации «bb».

**Задача №1-б.**

Алфавит {a,b}. Определить, есть ли в слове комбинация «ab». Если да, то каждую пару комбинацию «ab» заменить на комбинацию «bb». Если нет, то в конце слова написать символ «\*».

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово после разделителя «\*» в прямом порядке. При копировании удвоить все единички и пропустить все нули. Разделитель необходимо поставить самостоятельно.

**Задача №2-б.**

Алфавит {0,1}. Скопировать «скелет» слова после разделителя «\*» в обратном порядке. Разделитель необходимо поставить самостоятельно. Под «скелетом» понимается копия слова, в которой убраны все повторяющиеся символы. Например, вместо 000011100110 должно получиться 000011100110\*01010.

**Вариант 4.**

**Задача №1-а.**

Алфавит {1}. Дано число n, записанноев унарном коде. Число обязательно задано. Определить какое это число: если оно четное положительное, то поставить в конце слова «\*\*», если нечетное положительное – то поставить в конце слова «\*», если это число ноль – ничего не делать.

**Задача №1-б.**

Алфавит {1}. Дано число n, записанноев унарном коде. Число обязательно задано. Определить какое это число: если оно четное и больше трех, то поставить в конце слова «\*\*», если нечетное и больше трех – то поставить в конце слова «\*», если это число меньше или равно трем – стереть все слово и поставить вместо него «\*\*\*».

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово в прямом порядке без всякого разделителя.При копировании удвоить все «1».

**Задача №2-б.**

Алфавит {0,1}. Скопировать слово в прямом порядке без всякого разделителя.При копировании удвоить все нули и единицы, стоящие на четных позициях (первая позиция соотв. первому значимому символу слова).

***Методика оценки результатов выполнения***

***контрольной работы №1 (КР3) по курсу «ДМ (ТА и СВ)»***

**Задача 1.**

1. Полностью верный алгоритм – в версии «а» макс – 2 балла, в версии «б» - 3 балла.
2. Отсутствие комментариев – минус 0,5 балла.
3. Наличие некритичной ошибки (лишние инструкции, лишние состояния) – минус 0,5 балла за каждую ошибку.
4. Наличие критичной ошибки (отсутствие инструкции для возможной конфигурации, неправильная обработка крайних ситуаций) – минус 1 балл за каждую ошибку.
5. Неправильная логика работы программы – 0 баллов за всё задание.

**Задача 2**

1. Полностью верный алгоритм – в версии «а» макс – 3 балла, в версии «б» - 4 балла.
2. Отсутствие комментариев – минус 0,5 балла.
3. Наличие некритичной ошибки (лишние инструкции, лишние состояния) – минус 0,5 балла за каждую ошибку.
4. Наличие критичной ошибки (отсутствие инструкции для возможной конфигурации, неправильная обработка крайних ситуаций) – минус 1 балл за каждую ошибку.
5. Неправильная логика работы программы – 0 баллов за всё задание.

Максимальный балл за выполнение контрольной работы – **7 баллов**.

***Контрольная работа №2 (КР6)***

*Необходимо решить одну из двух задач №1 (типа «а» или «б») и одну из двух задач №2 (типа «а» или «б»). Решение следует прокомментировать.*

**Вариант 1.**

**Задача №1-а.**

Алфавит {0,1,2}. Определить, сколько в слове последовательностей «012». Результат записать после слова в унарном коде.

**Задача №1-б.**

Алфавит {0,1,2}. Определить, сколько в слове последовательностей «012». Если их четное количество, после слова поставить «+», если нечетное – то «-», если их нет – стереть всё слово.

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово после разделителя «\*» в прямом порядке. При копировании удвоить все нули. Разделитель необходимо поставить самостоятельно.

**Задача №2-б.**

Алфавит {0,1}. Скопировать «скелет» слова после разделителя «\*» в прямом порядке. Разделитель необходимо поставить самостоятельно. Под «скелетом» понимается копия слова, в которой убраны все повторяющиеся символы. Например, вместо 000011100110 должно получиться 000011100110\*01010.

**Вариант 2.**

**Задача №1-а.**

Алфавит {0,1,2}. Определить, сколько в слове последовательностей «012». Результат записать после слова в унарном коде.

**Задача №1-б.** Алфавит {0,1,2}. Определить, сколько в слове последовательностей «012». Если их четное количество, после слова поставить «+», если нечетное – то «-», если их нет – стереть всё слово.

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово после разделителя «\*» в прямом порядке. При копировании удвоить все нули. Разделитель необходимо поставить самостоятельно.

**Задача №2-б.**

Алфавит {0,1}. Скопировать «скелет» слова после разделителя «\*» в прямом порядке. Разделитель необходимо поставить самостоятельно. Под «скелетом» понимается копия слова, в которой убраны все повторяющиеся символы. Например, вместо 000011100110 должно получиться 000011100110\*01010.

**Вариант 3.**

**Задача №1-а.**

Алфавит {0,1}. Проанализировать слово: если  в нем  последовательность 010  
встречается только один раз переработать слово в 1, если два  раза  
- переработать слово в 2, если более двух раз - переработать слово в 3, если ни разу невстретилась переработать слово в 0**.**

**Задача №1-б.**

Алфавит {0,1}. Проанализировать слово: если  в нем  последовательность 010  
встречается четное число раз - переработать слово в 2, если нечетное число раз   
- переработать слово в 1, если ни разу невстретилась - не трогать слово и ничего не делать.

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово после разделителя «\*» в прямом порядке. При копировании удвоить все единички. Разделитель необходимо поставить самостоятельно.

**Задача №2-б.**

Алфавит {0,1}. Скопировать «скелет» слова после разделителя «\*» в обратном порядке. Под «скелетом» понимается копия слова, в которой убраны все повторяющиеся символы. Например, вместо 000011100110 должно получиться 000011100110\*01010.

**Вариант 4.**

**Задача №1-а.**

Проанализировать слово: если  в нем все нули одиночные -  
переработать в 1, если все нули встречаются только парами - в 2, если присутствуют иные варианты встречаемости нулей либо нулей вообще нет - в\***.**

**Задача №1-б.**

Проанализировать слово: если  в нем все нули одиночные -  
переработать в 1, если все нули встречаются только парами - в 2, если только тройками - в 3, если присутствуют иные варианты встречаемости нулей - в 4**,** если нулей нет- не трогать слово и ничего не делать**.**

**Задача №2-а.**

Алфавит {0,1}. Скопировать слово в прямом порядке без всякого разделителя.При копировании удвоить все «0».

**Задача №2-б.**

Алфавит {0,1}. Скопировать слово в **прямом** порядке без всякого разделителя.При копировании пропустить каждый второй встреченный ноль (символы отсчитываются начиная с **начала** слова).

***Методика оценки результатов выполнения***

***контрольной работы №2 (КР5) по курсу «ДМ (ТА и СВ)»***

**Задача 1.**

1. Полностью верный алгоритм – в версии «а» макс – 2 балла, в версии «б» - 3 балла.
2. Отсутствие комментариев – минус 0,5 балла.
3. Наличие некритичной ошибки (лишние инструкции, лишние состояния) – минус 0,5 балла за каждую ошибку.
4. Наличие критичной ошибки (отсутствие инструкции для возможной конфигурации, неправильная обработка крайних ситуаций) – минус 1 балл за каждую ошибку.
5. Неправильная логика работы программы – 0 баллов за всё задание.

**Задача 2**

1. Полностью верный алгоритм – в версии «а» макс – 3 балла, в версии «б» - 4 балла.
2. Отсутствие комментариев – минус 0,5 балла.
3. Наличие некритичной ошибки (лишние инструкции, лишние состояния) – минус 0,5 балла за каждую ошибку.
4. Наличие критичной ошибки (отсутствие инструкции для возможной конфигурации, неправильная обработка крайних ситуаций) – минус 1 балл за каждую ошибку.
5. Неправильная логика работы программы – 0 баллов за всё задание.

Максимальный балл за выполнение контрольной работы – **7 баллов**.

***Контрольная работа №3 (КР13)***

*Необходимо решить одну из двух задач №1 (типа «а» или «б») и одну из двух задач №2 (типа «а» или «б»). Решение следует прокомментировать.*

*Задание №1.* Рекурсию брать по переменной x. Строго доказать примитивную рекурсивность функции, получив **схему** примитивной рекурсии, в которую кроме базисных функций может входить функция суммы Σ и произведения П любого количества переменных, а также функция степени двух переменных St(x,y)=xy (первым указывается основание, вторым – показатель степени). *Ответ приводится полностью, без сносок и сокращений, в процессе решения сокращения возможны.*

*Задание №2.*Восстановить аналитический вид функций G и E. Найти значение функции *f(x)* в указанных точках. *Необходимы подробные пояснения при расчете значений функции*

**Вариант 1.**

**№1-а.**

**ϕ(x, y) =3(x+2y) +2(x+1)+3**

**№1-б.**

**ϕ(x, y,z) =3(x2+2y+1) +2yz+2(x+z+1)+3**

**№2-а.** *(Точки: x=40, x=93, x=202)*



*E(x) = *

*f(x)=*

**№2-б.** *(Точки: x=12, х=70, x=199)*



**Вариант 2.**

**№1-а.**

**ϕ(x, y) =y(x+3) +(x+1)у+2**

**№1-б.**

**ϕ(x, y,z) = (x+у)2 +xуz+zy+2**

**№2-а.** *(Точки: x=0, х=70, x=120)*





*f(x)=*

**№2-б.** *(Точки: x=0, х=30, x=55)*



*f(x)=*

**Вариант 3.**

**№1-а.**

**ϕ(x, y) =y(2x+1) +4+3(x+у)**

**№1-б.**

**ϕ(x, y,z) = (2x+1)2 +уz+4+(x+у+1)z2**

**№2-а.** *(Точки: x=19, x=47, x=200)*

*E*(*x*) = *R*5[*S*22(,*U*12, *C*22]

*G*(*x*) = *R*1[*S*22(П,*U*12, *C*22]

*f*(*x*) = *y*[*E*(*y*) + *G*(*y*) = *х*]

**№2-б.** *(Точки: x=10, x=144, x=529)*

*E*(*x*) = *R*6[*S*22(,*U*12, *C*12]

*G*(*x*) = *R*5[*S*22(П,*U*12, *C*22]

*f*(*x*) = *y*[*E*(*y+1*) + *x* = *G*(*y+1*)]

**Вариант 4.**

**№1-а.**

**ϕ(x, y) =3+2(2x+y) +2(x+1)y**

**№1-б.**

**ϕ(x, y,z) =3+2(2x2+2y) +2xy+z2**

**№2-а.** *(Точки: x=78, x=319, x=516)*

*E(x) = *

*G*(*x*) = *R*2[*S*22(П,*U*12, *C*32]

*f*(*x*) = *y*[*E*(*y*) + *G*(*y*) = *х*]

**№2-б. (***Точки: x=6, x=14, x=16)*





***Методика оценки результатов выполнения***

***контрольной работы №3 (КР13) по курсу «ДМ (ТА и СВ)»***

**Задача 1.**

1. Полностью верная схема рекурсии – в версии «а» макс – 2 балла, в версии «б» - 3 балла.
2. Наличие теоретической ошибки (неправильные верхние или нижние индексы) – минус 0,5 балла за каждую ошибку.
3. Наличие формальной ошибки (оставление функции вне суперпозиции) – минус 0,5 балла за каждую ошибку.
4. Наличие арифметической ошибки (неправильное выделение функции в предыдущей точке) – минус 1 балл.
5. Неправильная логика построения схемы рекурсии – 0 баллов за всё задание.

**Задача 2**

1. Полностью верное решение – в версии «а» макс – 3 балла, в версии «б» - 4 балла.
2. Неверно восстановленный аналитический вид функции G(x) - минус 1 балл.
3. Неверно восстановленный аналитический вид функции E(x) - минус 1 балл.
4. Наличие ошибки при подсчете функции f(x) в одной точке – минус 0,5 балла
5. Отсутствие комментариев – минус 0,5 балла.
6. Неправильная логика вычислений – минус 0,5 балла.

Максимальный балл за выполнение контрольной работы – **7 баллов**.

**Комплект тестовых заданий по дисциплине**

**Дискретная математика (Теория алгоритмов и сложность вычислений)**

***Тест №1 (Т-6)***

Тест №1 оформлен в виде комбинированного теста и включает в себя четыре группы вопросов

1. Группа вопросов – результат автоматической генерации вопросов из базы данных. Проверяется автоматически, оценка максимальная 1 балл.
2. Две теоремы из Списка теорем и Группа из трех определений из Списка определений. Проверяется вручную, оценка максимальная 3 балла.

***Тест №2 (Т-11)***

Тест №2 оформлен в виде комбинированного теста и включает в себя пять групп вопросов

1. Группа вопросов – результат автоматической генерации вопросов из базы данных. Проверяется автоматически, оценка максимальная 6 баллов.
2. Две теоремы из Списка теорем и Группа из трех определений из Списка определений. Проверяется вручную, оценка максимальная 3 балла.

***Тест №3 (Т-16)***

Тест №3 оформлен в виде комбинированного теста и включает в себя пять групп вопросов

1. Группа вопросов – результат автоматической генерации вопросов из базы данных. Проверяется автоматически, оценка максимальная 3 балла.
2. Две теоремы из Списка теорем и Группа из трех определений из Списка определений. Проверяется вручную, оценка максимальная 3 балла.

**Комплект материалов для оценивания работы на семинарских занятиях по дисциплине**

**Дискретная математика (Теория алгоритмов и сложность вычислений)**

На семинарские занятия выносятся задачи, которые с одной стороны вызывают наибольшие трудности и с другой стороны носят установочный, наводящий характер для освоения методики решения задач по курсу в целом.

**Задачи к разделу 1.**

**Задача 1.1**

Дано слово в алфавите {0,1}. Написать ''1'' в конце слова.

**Задача 1.2**

Дано слово в алфавите {0,1}. Написать в конце слова последний символ.

**Задача 1.3**

Дано слово в алфавите {0,1}. Скопировать 1-й символ, записать его в конце слова.

**Задача 1.4**

Дано слово в алфавите {0,1,2}. Все последовательности «012» заменить на последовательность из трех звездочек «\*\*\*».

**Задача 1.5**

Дано число n в унарном коде. Найти значение функции псевдоразности f(n)= n÷2. Результат записать вместо слова.

n-2, если n≥2

n÷2=

0, если n<2

Унарный код оперирует символами только одного типа, поэтому количество предметов кодируется поштучно, например ׀ (одна палочка) это число “0”, ׀׀ (две палочки) –число “1”, ׀׀׀ - “2”, и т.д.

**Задача 1.6**

Дано слово в алфавите {a,b}. Скопировать его в обратном порядке, записав копию сразу вслед за исходным словом.

**Задача 1.7**

Дано слово в алфавите {a,b}. Скопировать его в прямом порядке, записав копию сразу за исходным словом.

**Задача 1.8**

Дано слово в алфавите {a,b}. Скопировать его в обратном порядке, записав копию вместо исходного слова.

**Задача 1.9**

Дано слово в алфавите {a,b}. Определить, является ли слово палиндромом. Палиндром – слово, которое читается одинаково справа налево и слева направо. Если да – написать после слова «+», если нет – «-».

З**адача 1.10**

На ленте в унарном коде записаны числа x и y, между ними знак - .

Вместо исходной последовательности символов написать результат вычисления функции f(x,y)=x-y.

**Задача 1.11**

Алфавит {0,1}. Машина двухленточная. Дано слово, оно записано на первой ленте. Скопировать это слово на вторую ленту в обратном порядке.

**Задача 1.12**

Алфавит {0,1}. Машина трехленточная. Даны два числа в двоичном коде, они записаны на первой и второй ленте соответственно. Считается, что слова записаны корректно: т.е. оба слово всегда есть и начинаются они с единицы (исключение только для числа ноль). Определить чему равна сумма этих двух чисел и результат записать на третью ленту также в двоичном коде (и также корректно).

***Важно***: в ходе сложения, возможно, выяснится, что разрядов не хватает – тогда надо будет сдвинуть результат на одну клетку вправо для записи первой единицы.

**Задача 1.13**

Алфавит унарный. Машина трехленточная. Даны два числа в унарном коде, они записаны на первой и второй ленте соотв. Определить, чему равно произведение этих двух чисел и результат записать на третью ленту.

**Задача 1.14**

Алфавит {0,1}. Переработать произвольное слово в 0.

**Задача 1.15**

Алфавит {0,1}. Удвоить все символы в слове, результат записать вместо слова.

**Задача 1.16**

Алфавит {0,1}. Удалить каждый третий символ.

**Задача 1.17**

Алфавит {0,1}. Поставить в конце слова +, если в нем присутствует хотя бы один ноль и – , если это не так.

**Задача 1.18**

Алфавит {0,1}. Поставить в конце слова +, если в нем есть строго два нуля (не важно, соседние они или нет) и поставить в конце – , если это не так.

**Задача 1.19**

Алфавит {0,1}. Поставить в конце слова +, если в нем есть точно два нуля и они стоят рядом (т.е. они соседние) и в – иначе.

**Задача 1.20**

Алфавит {0,1}. Скопировать слово в прямом порядке после слова.

**Задача 1.21**

Даны два числа в унарном коде, разделенные знаком “+”. Вычислить сумму. Результат записать вместо слова.

**Задача 1.22**

Даны два числа в унарном коде, разделенные знаком “-”. Вычислить разность. Результат записать вместо слова.

**Задача 1.23**

Алфавит {a,b}. Подсчитать количество символов a. Результат записать после слова в унарном коде.

**Задача 1.24**

Даны два числа в унарном коде, разделенные знаком “/”. Вычислить целую часть от деления первого числа на второе. Результат записать вместо слова.

**Задача 1.25**

Алфавит {0,1}. Определить – каких символов 0 или 1 в слове больше. Если больше нулей – поставить в конце +, если больше 1 – поставить -. Если поровну – поставить в конце =.

**Задачи к разделу 2**

Установить взаимно-однозначное соответствие между промежутком a≤х≤b и c≤х≤d.

Установить взаимно-однозначное соответствие между промежутком 0≤х<1 и 0≤x<∞.

Установить взаимно-однозначное соответствие между промежутком -1<х<1 и 0≤x<∞.

Установить взаимно-однозначное соответствие между промежутком -1<х≤1 и 0<x<∞.

Установить взаимно-однозначное соответствие между промежутком 0≤х≤1 и множеством иррациональных чисел того же отрезка.

Установить взаимно-однозначное соответствие между всеми точками плоскости и точками сферы радиуса 1.

Установить взаимно-однозначное соответствие между точками открытого квадрата 0<х<1, 0<y<1 и точками плоскости.

Установить взаимно-однозначное соответствие между множеством всех рациональных чисел отрезка 0≤х≤1 и множеством всех точек плоскости, обе координаты которых рациональны.

Установить взаимно-однозначное соответствие между точками окружности радиуса a и b, a≥b.

Является ли счетным множество всех конечных подмножеств счетного множества?

Является ли счетным любое множество попарно непересекающихся букв Т на плоскости?

Является ли счетным любое множество попарно непересекающихся букв Г на плоскости?

Является ли счетным любое множество попарно непересекающихся открытых интервалов на действительной прямой?

Является ли счетным множество точек разрыва монотонной функции на действительной прямой?

Можно ли построить на плоскости несчетное множество попарно непересекающихся окружностей?

Можно ли построить на плоскости несчетное множество попарно непересекающихся букв N?

Можно ли построить на плоскости несчетное множество попарно непересекающихся букв Б?

Найти мощность множества всех четырехугольников на плоскости, координаты всех вершин которых рациональны.

Найти мощность множества всех многоугольников на плоскости, координаты всех вершин которых рациональны.



Найти мощность множества всех многочленов с действительными коэффициентами.



Найти мощность множества всех действительных чисел, в десятичном разложении которых встречается цифра 3.



Найти мощность множества всех действительных чисел, в десятичном разложении которых не встречается цифра 6.



Найти мощность множества всех действительных чисел 0<х<1, в десятичном разложении которых после запятой стоит цифра 4 и больше эта цифра не встречается.

Найти мощность множества всех непрерывных функций на действительной прямой.

**Условие для задач 3.1.-3.6.**

Пусть задана частичная арифметическая функция *ƒ(x).* Множество А- её область определенности, а множество В – её область значений. Найти характеристические функции множеств А и В.

**Задача 3.1.**

ƒ(x) = x–15.

**Задача 3.2.**

ƒ(x) = 2x+5.

**Задача 3.3.**

ƒ(x) = 3x–5.

**Задача 3.4.**

ƒ(x) = x – 5.

**Задача 3.5.**

ƒ(x) =(x-6) / (9-x).

**Задача 3.6**

ƒ(x) = 3x+4.

**Условие для задач 3.7.-3.12.**

Пусть задана функция ƒ от одного или двух аргументов. Укажите все множества (АФ, ВАФ, ЧАФ, ВЧАФ), к которым принадлежит функция ƒ.

**Задача 3.7.**

ƒ(x, y) = -x-1-y.

**Задача 3.8.**

ƒ(x, y) = 2x-1+3y.

**Задача 3.9**

ƒ(x) = 2x+3.

**Задача 3.10**

x+4, если машина Тх не остановится на чистой ленте;

ƒ(x) = 1, если остановится за первые 15 тактов;

3x-1, если остановится позднее 15-ого такта.

**Задача 3.11**

1, если машина Тх остановится за первые 18 тактов;

ƒ(x) =

x+2, если машина Тх остановится позднее 18-ого такта.

**Задача 3.12**

3x-2y, если x > y;

ƒ(x, y) = y-x, если x < y;

(x+y)/2 если x=y.

**Условие для задач 3.13.-3.15.**

Для некоторой частичной арифметической функции ƒ(x) заданы множество А (область определенности) и множество В (область значений). Найти аналитический вид функции ƒ(x) и задать множества A и B с помощью характеристических функций.

**Задача 3.13**

А: (x≥11) ∩ (mod(x,2)=0);

В: N.

**Задачи к разделу 3.**

**Задача 4.1.**

Доказать примитивную рекурсивность функции суммы

*f(x,y)=x+y.*

**Задача 4.2.**

Доказать примитивную рекурсивность функции произведения двух аргументов: *f(x,y)=П(x,y)=x∙y.*

**Задача 4.3.**

Доказать примитивную рекурсивность функции факториал

**Задача 4.4.**

Доказать примитивную рекурсивность функции псевдоразности:

*x-y,* если *x≥y;*

*f(x,y)= xy =*

*0,* иначе.

**Задача 4.5.**

Доказать примитивную рекурсивность знаковой функции:

*0,* если *x=0;*

*Sg (x) =*

*1,* если *x>0.*

**Задача 4.6.**

Доказать примитивную рекурсивность функции равенства:

*1,* если *x = y;*

*Eql (x,y) =*

*0,* иначе.

**Задача 4.7.**

Доказать примитивную рекурсивность функции степени:

*f(x,y)=xy .*

**Задача 4.8.**

Доказать примитивную рекурсивность функции модуля разности:

*x y,* если *x≥y;*

*Mod (x,y)=*

*y x,* иначе.

**Задача 4.9.**

Доказать примитивную рекурсивность функции больше:

*1,* если *x>y;*

*More (x,y) =*

*0,* иначе.

**Задача 4.10.**

Доказать примитивную рекурсивность функции больше или равно:

*1,* если *x≥y;*

*Moreql (x,y) =*

*0,* иначе.

**Задача 4.11.**

Доказать примитивную рекурсивность функции остаток от деления аргумента *x* на 2:

*rest2(x).*

**Задача 4.12.**

Доказать примитивную рекурсивность функции минимум:



**Задача 4.13.**

Доказать примитивную рекурсивность функции максимум:



**Задача 4.14.**

Доказать примитивную рекурсивность функции целая часть от деления аргумента *x* на 2:

*div2(x).*

**Задача 4.15.**

Доказать примитивную рекурсивность функции целая часть от деления аргумента *x* на *y*.

*div (x,y)*

**Задача 4.16.**

Доказать примитивную рекурсивность функции остаток от деления аргумента *x* на *y:*

*rest (x,y).*

**Задача 4.17.**

Доказать примитивную рекурсивность функции число различных делителей *x* (включая число 1):

*nd (x).*

**Задача 4.18.**

Доказать примитивную рекурсивность функции *(n+1)*-е простое число в натуральном ряде чисел:

*p(n).*

***Методика оценки результатов работы***

*на семинаре по курсу*

***Дискретная математика (Теория алгоритмов и сложность вычислений)***

Оценка знаний студента на семинарском занятии преподавателем определяется следующими основными критериями: активность и посещаемость.

Сем (баллы за работу на семинаре) – дополнительные баллы за каждый раздел, выставляются студенту за посещение семинарских занятий и проявленную на нах активность.

Правила выставления баллов за посещаемость семинарских занятий (по разделам)

• Нет пропусков или не более одного пропуска +1 балл

• Два и более пропуска - 0 баллов

Правила выставления баллов за активность на семинарских занятий (по разделам)

• За каждый ответ на семинаре (решение задачи у доски или самое быстрое в группе решение в своей тетради) + 0,5 балла

Максимальный балл за работу на семинаре в рамках раздела – 3 балла.

**Вопросы к экзамену по дисциплине**

**Дискретная математика (Теория алгоритмов и сложность вычислений)**

Экзаменационный билет оформлен в виде сквозного (по всем темам курса) комбинированного теста и включает в себя:

1. Две теоремы из Списка теорем.
2. Пять определений из Списка определений.
3. Набор вопросов – результат автоматической генерации вопросов из базы данных.

**Список теорем**

|  |  |
| --- | --- |
| **Название** | **Формулировка (надо знать обязательно, кроме раздела «теоремы» может быть проверена в разделе «определения»)** |
|  | Всякая k -ленточная машина Тьюринга М может быть преобразована в эквивалентную машину М\* с одной лентой. |
| **Теорема Шеннона №1** | Всякая машина Тьюринга *А* может быть преобразована в эквивалентную машину *В* не более чем с двумя внутренними состояниями. |
| **Теорема Шеннона №2** | Всякая машина Тьюринга *А* может быть преобразована в эквивалентную машину *С* не более чем с двумя знаками внешнего алфавита. |
|  | Задача об остановке произвольной машины Тьюринга на произвольном входном слове алгоритмически неразрешима. |
|  | Задача об остановке произвольной машины Тьюринга на пустой ленте алгоритмически неразрешима |
|  | Задача о печатании данного символа на чистой ленте точно один раз алгоритмически неразрешима |
|  | Задача о печатании данного символа на чистой ленте бесконечно много раз алгоритмически неразрешима |
|  | Задача о печатании данного символа на чистой ленте хотя бы один раз алгоритмически неразрешима |
| **Теорема Райса** | Ни для ка­ко­го не­три­ви­аль­но­го ин­ва­ри­ант­но­го свой­ст­ва ма­шин Тью­рин­га не су­ще­ст­ву­ет алгоритма, по­зво­ля­юще­го для лю­бой ма­ши­ны Тьюрин­га уз­нать, об­ла­да­ет ли она этим свой­ст­вом |
|  | Множество машин Тьюринга эффективно перечислимо |
|  | Множество алгоритмов Маркова эффективно перечислимо |
| **Теорема Поста** | Если множество А эффективно перечислимо, то подмножество В эффективно распознается в А тогда и только тогда, когда В и А\В оба эффективно перечислимы |
|  | Множество останавливающихся машин Тьюринга эффективно перечислимо |
|  | Множество не останавливающихся машин Тьюринга невозможно эффективно перечислить |
| **Парадокс Галилея** | Хотя большинство натуральных чисел не является квадратами, всех натуральных чисел не больше, чем квадратов (если сравнивать эти множества по мощности) |
| **Парадокс Гильберта** | Если гостиница с бесконечным количеством номеров полностью заполнена, в неё можно поселить ещё посетителей, даже бесконечное число |
|  | Множество целых чисел счетно и эффективно перечислимо |
|  | Множество упорядоченных пар натуральных чисел счетно и эффективно перечислимо |
|  | Множество упорядоченных n-ок натуральных чисел счетно и эффективно перечислимо |
|  | Множество конечных комплексов натуральных чисел счетно и эффективно перечислимо |
|  | Множество рациональных чисел счетно и эффективно перечислимо |
|  | Множество алгебраических чисел счетно и эффективно перечислимо |
|  | Множество элементов, которые можно представить с помощью конечного числа счетной системы знаков, счетно |
|  | Множество действительных чисел несчётно |
|  | Множество комплексных чисел несчетно |
|  | Множество иррациональных чисел несчетно |
|  | Множество трансцендентных чисел несчетно |
|  | Между любыми двумя различными рациональными числами всегда найдется множество иррациональных чисел мощности континуума |
|  | Между любыми двумя различными иррациональными числами всегда найдется счетное множество рациональных чисел |
| **Теорема Кантора** | Для любого кардинального числа α справедливо α<2α |
|  | Для любого множества А найдется множество В, мощность которого больше А |
| **Парадокс Кантора** | Кардинальное число множества всех подмножеств P(U) множества всех множеств U не больше чем |U| |
| **Парадокс Рассела** | Пусть В – множество всех множеств, которые не содержат самих себя в качестве своих собственных элементов. Тогда можно доказать две теоремы: В принадлежит В и В не принадлежит В |
|  | Множество вычислимых действительных чисел счетно |
|  | Существуют невычислимые действительные числа и их несчетное множество |
|  | Множество арифметических функций n-переменных несчетно |
|  | Множество вычислимых арифметических функций счетно |
| **Теорема Тьюринга** | Множество вычислимых арифметических функций *n* переменных не поддается эффективному перечислению |
|  | Множество невычислимых арифметических функций несчетно |
|  | Множество арифметических функций, описываемых конечным числом слов, счетно и эффективно перечислимо |
|  | Множество частичных арифметических функций несчетно |
|  | Множество вычислимых частичных арифметических функций счетно и эффективно перечислимо |
|  | Невозможно эффективно распознать функции-константы среди вычислимых арифметических функций |
|  | Вычислимые арифметические функции не поддаются эффективному сравнению |
|  | Невозможно эффективно распознать функции-тождества среди вычислимых арифметических функций |
|  | Невозможно эффективно распознать вычислимые арифметические функции среди вычислимых частичных арифметических функций |
| **Теорема Черча** | Невозможно эффективно распознать точки неопределенности вычислимой частичной арифметической функции |
| **Теорема о мажорируемых неявных функциях** | Пусть *g(x1,…,xn, y), a(x1,…,xn)* – такие примитивно-рекурсивные функции, что уравнение *g(x1,…,xn, y)=0*  для каждых *x1,…,xn* имеет по меньшей мере одно решение и *μy(g(x1,…,xn, y)=0) ≤ a(x1,…,xn)* для любых *x1,…,xn*. Тогда функция *f(x1,…,xn)= μy(g(x1,…,xn, y)=0)* также примитивно рекурсивна |
|  | Примитивно-рекурсивные функции эффективно перечислимы |
|  | Множество частично-рекурсивных функций эффективно перечислимо |
|  | Общерекурсивные функции не поддаются эффективному перечислению |
|  | Примитивно-рекурсивные функции не поддаются эффективному распознаванию среди общерекурсивных функций |
|  | Общерекурсивные функции не поддаются эффективному распознаванию среди частично рекурсивных функций |

**Список определений**

|  |  |
| --- | --- |
| Термин | Определение |
| Алгоритм | это точное предписание о выполнении в некотором порядке системы операций, определяющих процесс перехода от исходных данных к искомому результату для решения задачи данного типа. |
| Машина Тьюринга | абстрактная (воображаемая) "вычислительная машина" некоторого точно охарактеризованного типа, дающая пригодное для целей математического рассмотрения уточнение общего интуитивного представления об алгоритме. |
| Управляющая головка | это некоторое устройство, которое может перемещаться вдоль ленты так, что в каждый рассматриваемый момент времени оно находится напротив определенной ячейки ленты |
| Внутренняя память машины | это выделенная ячейка памяти, которая в каждый рассматриваемый момент находится в некотором «состоянии». |
| Конфигурация машины Тьюринга | это совокупность, образованная содержимым текущей обозреваемой ячейки aj и состоянием внутренней памяти Si. |
| Программа машины Тьюринга | это совокупность команд установленного формата. |
| Тезис Тьюринга | любой алгоритм можно преобразовать в машину Тьюринга |
| Машина В эквивалентна машине А | если в соответствующие такты их работы лента машины В содержит всю информацию о ленте машины А. |
| Самоанализирующие машины | это машины, в которых все служебные символы как-либо изображаются ленточными символами. |
| Универсальная машина | машина Тьюринга, обладающая способностью путём подходящего кодирования выполнить любое вычисление |
| Взаимозаменяемые машины Тьюринга | Две ма­ши­ны Тью­рин­га, име­ющие один и тот же внешний ал­фа­вит, бу­дем на­зы­вать взаимозаменяемыми, если, ка­ко­во бы ни бы­ло сло­во в их об­щем ал­фа­ви­те, они ли­бо пе­ре­ра­ба­ты­ва­ют его в од­но и то же сло­во, ли­бо обе к не­му не­при­ме­ни­мы. |
| Свойст­во ма­шин Тью­рин­га на­зы­ва­ет­ся инвариантным | ес­ли лю­бые две взаимозаменяемые ма­ши­ны ли­бо обе об­ла­да­ют этим свой­ст­вом, ли­бо обе не об­ла­да­ют. |
| Свой­ст­во ма­шин Тью­рин­га на­зы­ва­ет­ся нетривиальным | ес­ли су­ще­ст­ву­ют как ма­ши­ны, обладающие этим свой­ст­вом, так и не об­ла­дающие им |
| Нормальный алгоритм Маркова | математическое построение, предназначенное для уточнения понятия алгоритм, которое задается алфавитом и нормальной схемой подстановок, выполняемых по заранее определенной схеме. |
| Тезис Маркова | любой вычислительный процесс можно преобразовать в нормальный алгоритм. |
| Эффективно перечислимое множество | множество, элементы которого можно перечислить по алгоритму (пронумеровать натуральным рядом без пропусков и повторений). |
| Подмножество В множества А эффективно распознается в А | если существует алгоритм, позволяющий однозначно для каждого элемента множества А определить, принадлежит ли данный элемент множеству В или дополнению В до А. |
| Множество (по Тьюрингу) | это объединение в одно общее объектов, хорошо различимых нашей интуицией или нашей мыслью. |
| Множество (по Кантору) | это совокупность объектов безразлично какой природы, неизвестно существующих ли, рассматриваемая как единое целое |
| Мощность множества (по Кантору) | это та общая идея, которая остается у нас, когда мы, мысля об этом множестве, отвлекаемся как от всех свойств его элементов, так и от их порядка. |
| Мощность множества | это характеристика, которая объединяет данное множество с другими множествами, применение процедуры сравнения к которым дает основание предполагать, что каждый элемент одного множества имеет парный элемент из другого множества и наоборот. |
| Конечное множество | множество, состоящее из конечного числа элементов, его кардинальное число совпадает с одним из натуральных чисел. |
| Бесконечное множество | множество, состоящее из бесконечного числа элементов, его кардинальное число совпадает с одним из трансфинитных чисел. |
| Счетно-бесконечные множества | бесконечные множества, равномощные множеству натуральных чисел (их элементы можно пронумеровать натуральными числами без пропусков и повторений). |
| Несчетные множества | бесконечные множества, не равномощные множеству натуральных чисел |
| Счетное множество | множество, являющееся конечным или счетно-бесконечным |
| Трансфинитное число | кардинальное число бесконечного множества |
| Множество целых чисел | множество, состоящее из натуральных чисел, числа ноль и чисел, построенных на основе натуральных только со знаком «минус» (отрицательных чисел). |
| Два элемента a и b называют упорядоченной парой | если указано, какой их этих элементов первый, а какой второй и при этом ((a,b)=(c,d))<=>(a=c)^(b=d). Упорядоченную пару элементов обозначают (a,b). |
| Упорядоченная n-ка натуральных чисел | это набор из n элементов вида (m1, m2, …, mn), где mi – натуральное число. |
| Конечные комплексы натуральных чисел | это элементы вида (p1), (p1, p2), (p1, p2, p3), …, (p1, p2, …, pk), где k и pi пробегают все натуральные числа. |
| Рациональное число | это число вида , где n – целое число, m – натуральное число. |
| Алгебраическое действительное число | это действительный корень алгебраического уравнения ненулевой степени с рациональными коэффициентами |
| Континуум-гипотеза | с точностью до эквивалентности, существуют только два типа бесконечных числовых множеств: счетное множество и континуум. |
| Комплексное число | задается парой (r1, r2), где r1, r2 принадлежат множеству действительных чисел |
| Иррациональное число | это действительное число, не являющееся рациональным. |
| Трансцендентное число | это действительное число, не являющееся алгебраическим. |
| Импредикабельное свойство | это свойство, которое не применимо само к себе. |
| Действительное число вычислимо | если существует алгоритм его вычисления с любой степенью точности |
| Арифметическая функция | это функция, определенная на расширенном множестве натуральных чисел N\* и принимающая значения из расширенного множества натуральных чисел N\*.. |
| Частичная арифметическая функция (ЧАФ) | это функция, определенная на некотором подмножестве М расширенного множества натуральных чисел N\* и принимающая значения из множества N\*. |
| Характеристическая функция χA какого-нибудь подмножества А расширенного множества натуральных чисел N\* | это функция от одной переменной, равная 1 в точках множества A и равная 0 в точках, не принадлежащих A |
| Вычислимая арифметическая функция (ВАФ) | это функция, для которой существует алгоритм вычисления ее значения в любой точке расширенного множества натуральных чисел N\* |
| Вычислимая частичная арифметическая функция (ВЧАФ) | это функция, для которой существует алгоритм вычисления ее значения в любой точке области определенности |
| Эффективное распознавание функций | это процедура, позволяющая при помощи некоторого алгоритма определить, относится ли данная функция к рассматриваемому классу |
| Эффективное сравнение арифметических функций | это процедура, позволяющая при помощи некоторого алгоритма определить, совпадают ли значения функций во всех точках |
| Частичная арифметическая функция f называется примитивно-рекурсивной | если она может быть получена из простейших функций Cqn, S, Umn конечным числом операций подстановки и примитивной рекурсии (т.е. задана в базисе Клини). |
| Частичная арифметическая функция f называется частично-рекурсивной | если она может быть получена из простейших функций Cqn, S, Umn конечным числом операций подстановки, примитивной рекурсии и минимизации (т.е. задана в расширенном базисе Клини). |
| Тезис Черча | Класс алгоритмически (или машинно) вычислимых частичных арифметических функций совпадает с классом всех частично рекурсивных функций. |
| Частичная арифметическая функция f называется общерекурсивной | если она может быть получена из из простейших функций Cqn, S, Umn конечным числом операций подстановки, примитивной рекурсии и слабой минимизации |
| Кусочно-заданная функция | Пусть заданы некоторые функции fi(x1,.....,xn), i=1,…,s+1 и указаны какие то условия Pj(x1,.....,xn), j=1,…,s, которые для любого набора чисел x1,.....,xn могут быть истинными или ложными. Допустим, что ни для одного набора чисел x1,.....,xn никакие два из упомянутых условий не могут быть одновременно истинными. Функция f(x1,.....,xn), заданная схемой:  f1(x1,…,xn) если P1(x1,…,xn) истинно;  f2(x1,…,xn) если P2(x1,…,xn) истинно;  f (x1,.....,xn) = ………  fs(x1,…,xn) если Ps(x1,…,xn) истинно;  fs+1(x1,…,xn) для остальных x1,…,xn  называется кусочно-заданной |
| Нерекурсивная функция | функция, значение которой нельзя вычислить, несмотря на то, что в данной рассматриваемой точке сама функция определена |
| Непримитивно рекурсивная функция | функция, являющаяся общерекурсивной, но не принадлежащая к множеству примитивно-рекурсивных функций. |
| Временная сложность алгоритма T(x) | это число его шагов, необходимых для решения данной задачи размера x в худшем случае. |
| Пространственная сложность S(x) | это число единиц памяти, необходимых для решения задачи размера x в худшем случае. |
| Арифметическая функция f(x) называется функцией одного верхнего порядка с функцией g(x) и записывается … | f(x) = O(g(x)), если существует такое натуральное число c > 0, что, в конце концов (т. е. начиная с некоторого x) получим f(x)  c∙g(x). |
| Арифметическая функция f(x) называется функцией одного нижнего порядка с функцией g(x) и записывается … | f(x)=о(g(x)), если существует такое натуральное число c > 0, что, в конце концов (начиная с некоторого x) получим f(x) ≥ c∙g(x) |
| Арифметическая функция f(x) называется функцией одного порядка с функцией g(x) и записывается … | f(x) = o(g(x)) & f(x) = О(g(x)), если она одного верхнего и одного нижнего порядка с функцией g(x) |
| Полиномиальные функции | это функции одного верхнего порядка с многочленами |
| Экспоненциальные функции | это функции одного нижнего порядка с экспонентой |
| Арифметические функции f(x) и g(x) называются полиномиально-связанными или полиномиально-эквивалентнтными | если существуют такие многочлены P1(x) и P2(x), что, в конце концов (начиная с некоторого числа) f(x)≤P1∙g(x) и g(x)≤P2∙f(x) |
| Полиномиальный алгоритм | это алгоритм, обе функции сложности которого полиномиальные, |
| Экспоненциальный алгоритм | это алгоритм, у которого хотя бы одна из двух функций сложности экспоненциальная |
| Легкоразрешимая задача | это задача, решаемая полиномиальным алгоритмом |
| Трудноразрешимая задача | это задача, которую нельзя решить полиномиальным алгоритмом |

***Методика оценки результатов экзамена***

*по курсу*

***Дискретная математика (Теория алгоритмов и сложность вычислений)***

Максимальный балл за экзамен – 50 баллов. Из них 25 баллов можно получить за устную часть и 25 баллов за автоматизированную. Оценка выставляется пропорционально сложности заданных вопросов и корректности/полноты ответов на них.

**Итоговая оценка по курсу выставляется в соответствии**

**со следующей таблицей:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сумма баллов по дисциплине** | **Оценка по 4-х бальной шкале** | **Зачет** | **Оценка (ECTS)** | **Градация** |
| 90 - 100 | 5 (отлично) | Зачтено | А | Отлично |
| 85 - 89 | 4 (хорошо) | В | Очень хорошо |
| 75 - 84 | С | Хорошо |
| 70 - 74 | 3 (удовлетворительно) | D | Удовлетворительно |
| 65 - 69 |
| 60 - 64 | E | Посредственно |
| Ниже 60 | 2 (неудовлетворительно) | Не зачтено | F | Неудовлетворительно |

Автор(ы):

|  |  |  |
| --- | --- | --- |
|  | Тихомирова Анна Николаевна, к.т.н. |  |