

УТВЕРЖДЕНО  
Проректор по учебной работе  
и довузовской подготовке  
А. А. Воронов  
27 июня 2017 г.

## ПРОГРАММА

по дисциплине: **ИНФОРМАТИКА. Основной курс**  
по направлению подготовки: **03.03.01 «Прикладные математика и физика»**  
физтех-школа: **ФАКТ**  
факультет: **ФАКИ, ФАЛТ**  
кафедра: **информатики и вычислительной математики**  
курс: 1  
семестр: 1

Трудоемкость:

Базовая часть – 4 зачет. ед.:

лекции: – 30 часов

практические (семинарские)

занятия: – нет

лабораторные занятия: – 60 часов

Экзамен – нет

Зачет дифф. – 1 семестр

Две контрольные работы

Самостоятельная работа – 90 часов

ВСЕГО АУДИТОРНЫХ ЧАСОВ – 90

Программу составили: академик РАН В. П. Иванников,  
доцент, к.ф.-м.н. П. Н. Коротин  
доцент, к.т.н. В. К. Хохлов

Программа принята на заседании  
кафедры информатики и вычислительной математики  
16 июня 2017 г.

Заведующий кафедрой,  
чл.-корр. РАН

И. Б. Петров

Структура преподавания дисциплины.

**Введение в теорию алгоритмов.** Интуитивное понятие алгоритма. Свойства алгоритмов. Понятие об исполнителе алгоритма. Алгоритм как преобразование слов из заданного алфавита. Связь понятия алгоритма с понятием функции. Машина Тьюринга. Нормальные алгоритмы Маркова. Вычислимые функции и их свойства. Невычислимые функции. Различные эквивалентные определения множества вычислимых функций. Алгоритмическая сложность.

**Алгоритмические языки.** Характеристика алгоритмических языков и их исполнителей. Понятие трансляции.

Понятие о формальных языках. Способы строгого описания формальных языков, понятие о метаязыках. Алфавит, синтаксис и семантика алгоритмического языка. Описание синтаксиса языка с помощью металингвистических формул и синтаксических диаграмм.

Языки программирования. Общие характеристики языков программирования. Алфавит, имена, служебные слова, стандартные имена, числа, текстовые константы, разделители. Препроцессор и комментарии.

Типы данных, их классификация. Переменные и константы. Скалярные типы данных и операции над ними. Старшинство операций, стандартные функции. Выражения и правила их вычисления. Оператор присваивания.

Файлы. Стандартные функции ввода-вывода.

Простые и сложные операторы. Пустой, составной, условный операторы. Оператор варианта. Оператор перехода.

Оператор цикла. Программирование рекуррентных соотношений.

Составные типы данных. Массивы.

Описание функций (процедур). Формальные и фактические параметры. Способы передачи параметров. Локализация имен. Побочные эффекты. Итерации и рекурсии.

Ссылочный тип данных. Методы выделения памяти: статический, динамический и автоматический. Структуры. Битовые поля. Объединения. Перечисления. Декларация typedef.

**Алгоритмы сортировки.** Понятие внутренней и внешней сортировки. Устойчивая сортировка. Сортировка in-place. Сортировка простыми вставками, простым выбором, метод «пузырька». Шейкер сортировка. Метод Шелла. Быстрая сортировка Хоара. Сортировка слиянием. Пирамидальная сортировка. Оценка трудоемкости.

**Алгоритмы и структуры данных.** Абстрактные структуры данных: список, стек, очередь, очередь с приоритетом, ассоциативный массив. Отображение абстрактных структур данных на структуры хранения: массивы, линейные списки, деревья.

Различные реализации ассоциативного массива: двоичные деревья поиска, перемешанные таблицы (с прямой и открытой адресацией, использование техники двойного хэширования при открытой адресации). Оценки алгоритмической сложности операций поиска, добавления и удаления элемента.

Классические алгоритмы: перебор с возвратом, жадные алгоритмы.

## **Учебно-методическое и информационное обеспечение дисциплины**

### **Основная литература**

1. *Ворожцов А. В., Винокуров Н. А.* Практика и теория программирования. – М.: Физматкнига, 2008.
2. *Керниган Б. У., Ритчи Д. М.* Язык программирования С. – 2-е изд. – М.: Издательский дом «Вильямс», 2006.

### **Дополнительная литература**

1. *Шилдт Г.* Полный справочник по С. – М.: Издательский дом «Вильямс», 2005.
2. *Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К.* Алгоритмы: построение и анализ. – 3-е изд. – М.: ООО Издательский дом «Вильямс», 2013.
3. *Дасгупта С., Пападимитриу Х., Вазирани У.* Алгоритмы. Пер. с англ. под ред. А. Шеня. – М.: МЦНМО, 2014.
4. *Митницкий В. Я.* Элементы теории алгоритмов и язык программирования С. – М.: МФТИ, 2001.

### **Пособия и методические указания**

1. *Прут В. В.* Функции алгебры логики. Алгоритмы на языке С. Методические указания. – М.: МФТИ, 2013.
2. *Прут В. В.* Полные системы функций алгебры логики. Алгоритмы на языке С. Методические указания. – М.: МФТИ, 2013.
3. *Прут В. В.* Основные комбинаторные алгоритмы на языке С. Методические указания. – М.: МФТИ, 2013.
4. *Прут В. В.* Алгоритмы обработки структур данных на языке С. Списки, стеки, деревья. Методические указания. – М.: МФТИ, 2013.
5. *Прут В. В.* Сортировка и поиск. Алгоритмы на языке С. Учебно-методическое пособие. – М.: МФТИ, 2014.
6. *Прут В. В.* Введение в графы. Алгоритмы на языке С. Учебно-методическое пособие. – М.: МФТИ, 2014.
7. *Прут В. В.* Свойства графов. Алгоритмы на языке С. Учебно-методическое пособие. – М.: МФТИ, 2014.

### **Электронные ресурсы**

1. <http://cs.mipt.ru>
2. <http://acm.mipt.ru>

## ЗАДАНИЕ 1

(срок сдачи 23–28 октября)

### 1. Машины Тьюринга

Обозначим как  $N_0$  множество всех неотрицательных целых чисел.

Описать машины Тьюринга, которые реализуют:

1.1. Счетчик четности. Выход машины Тьюринга равен 0 или 1 в зависимости от того, четно или нечетно число единиц в последовательности из 0 и 1, записанной на ленте машины Тьюринга. В конце последовательности стоит символ  $B$ . В начальном состоянии головка видит первый левый символ.

1.2. Инверсию заданного слова в алфавите  $\{0, 1\}$  (0 заменяет на 1, а 1 – на 0).

1.3. «Переворачивание» заданного слова в алфавите  $\{a, b, c\}$ .

1.4. Сложение двух чисел из множества  $N_0$ , записанных на ленте в виде последовательности единиц, а именно:

$$0 \rightarrow 0, 1 \rightarrow 01, 2 \rightarrow 011, 3 \rightarrow 0111, 4 \rightarrow 01111, \dots$$

Назовём эту запись *единичной записью* числа. Записи чисел разделены на ленте несколькими пустыми ячейками. Рассмотрите различные варианты начального положения головки машины Тьюринга.

1.5. Сложение двух чисел из  $N_0$ , данных в двоичной системе счисления.

1.6. Распознавание правильных скобочных выражений. Правильное скобочное выражение – это слово в алфавите  $A = \{(\,)\}$ , которое может получиться, если из арифметического выражения удалить все символы, кроме скобок. Примеры правильных скобочных выражений: пустое слово,  $()$ ,  $((\ ))$ ,  $(\ )(\ )$ . Примеры неправильных скобочных выражений:  $(\ )(\ )(\ )$ ,  $(\ )\ )\ )\ )$ ,  $((\ ))$ . Результат работы: слово «YES», если скобочное выражение правильное, и слово «NO» – иначе.

### 2. Алгоритмы Маркова

2.1. Записать нормальные алгоритмы Маркова, которые реализуют:

2.1.1. Приписывание буквы  $X$  к входному слову справа.

2.1.2. Задание 1.2.

2.1.3. Задание 1.3.

2.1.4. Задание 1.5.

2.1.5. Задание 1.6.

2.1.6. Удвоение числа, заданного а) в виде единичной записи, б) в двоичной системе счисления.

2.2. В алфавите  $A = \{a, b\}$  описать нормальный алгоритм, который выдает в качестве результата пустое слово, если буквы  $a$  и  $b$  входят во входное слово в равном количестве, и любое непустое слово – иначе. В алгоритме должно быть не более четырех правил подстановки. Докажите правильность придуманного алгоритма.

### 3. Решение простых алгоритмических задач

В каждой задаче ответьте на вопрос о том, как растет время работы программы и используемая программой память с ростом параметра размера входных данных (например, параметра  $n$ ).

3.1. «Мах». Написать программу, которая выводит максимальное число из  $n$  заданных чисел. В первой строчке входа дано число  $n$ , а в следующей строчке указано  $n$  целых чисел.

3.2. «Числа Фибоначчи I». Написать программу, которая по данному  $n$  находит  $n$ -е число Фибоначчи  $F_n$ . Числа Фибоначчи определяются соотношениями

$$F_n = F_{n-1} + F_{n-2}, F_1 = F_2 = 1.$$

3.3. «Числа Фибоначчи II». Решить предыдущую задачу, используя идею рекурсии. Оценить число элементарных операций, которое необходимо сделать в рекурсивном и нерекурсивном алгоритмах вычисления числа  $F_n$ .

3.4. «Биномиальные коэффициенты». Написать программу, которая для данного натурального числа  $n$  находит коэффициенты в разложении

$$(1+x)^n = C_n^0 x^0 + C_n^1 x^1 + \dots + C_n^n x^n.$$

Использовать соотношения

$$C_n^n = C_n^0 = 1, C_n^k = C_{n-1}^k + C_{n-1}^{k-1}.$$

Оценить, как растет время работы вашей программы с ростом  $n$ .

3.5. «Простые числа». Написать программу, которая определяет, является ли введенное число  $n$  простым.

3.6. «Скобки». Написать программу, которая определяет, является ли введенная скобочная структура правильной (см. задание 1.6).

3.7. «Задача Иосифа». Пусть  $n$  человек, стоящие по кругу, считаются (начиная с первого) считалкой из  $m$  слов. Человек, на котором считалка заканчивается, – выбывает, круг смыкается, а счет продолжается с человека, следующего за выбывшим. Напишите программу, выводящую номера трех человек, выбывших последними, в порядке их выбывания. При написании программы следует использовать динамические переменные.

### ЗАДАНИЕ 2

(срок сдачи 4–9 декабря)

#### 1. Решение алгоритмических задач

В каждой задаче ответьте на вопрос о том, как растет время работы программы и используемая программой память с ростом параметра размера входных данных (например, параметра  $n$ ).

1.1. В строках текстового файла содержатся разделенные пробелом целочисленные координаты вершин прямоугольной односвязной плоской фигуры. Вершины фигуры перечисляются в произвольном порядке. Вычислить площадь заданной фигуры.

1.2. На поле СЗ шахматной доски неподвижно располагается белый король. Произвольно на доску выставляются черный король и белый ферзь. Напишите программу, доказывающую, что при любой (имеющей смысл) расстановке двух последних фигур мат черному королю может быть объявлен не более чем за 25 ходов.

## **2. Жадные алгоритмы**

2.1. «Атлеты». Написать программу, которая находит «башню» из атлетов максимальной высоты. Атлеты характеризуются двумя параметрами – массой и силой. Сила равна максимальной массе, которую атлет может держать на плечах. Известно, что если атлет тяжелее, то он точно сильнее. Подсказка: упорядочите атлетов по силе и стройте башню сверху. Вверх естественно поместить самого слабого. Входом является число атлетов  $n$  и  $n$  пар (масса, сила).

2.2. «Отрезки». Написать программу, которая для множества заданных отрезков находит минимальное подмножество отрезков, объединение которых покрывает отрезок  $S = [0, 10000]$ . Число отрезков и координаты их концов заданы на входе. Все координаты целочисленные. Подсказка: покрывайте отрезок  $S$  пошагово, двигаясь слева направо. На каждом шаге будет непокрытая часть  $[x, 10000]$ . Из оставшихся отрезков выбирайте тот, который урежет непокрытую часть до  $[y, 10000]$ , где  $y$  максимальное. Решите эту задачу за время  $O(n \log n)$ , где  $n$  – количество данных отрезков.

## **3. Структуры данных: списки, стек, очередь, деревья поиска и перемешанные таблицы (хэш-таблицы).**

1.1. «Скобки». Дано слово, состоящее из круглых и фигурных скобок. Написать программу, которая определяет, является ли введенное слово правильным скобочным выражением. Подсказка: постепенно считывайте скобки и используйте стек для хранения открывающих скобок, для которых пока не считаны парные закрывающие скобки. (Рекурсивное определение правильного скобочного выражения: слово называется правильным скобочным выражением, если все скобки в нем можно разбить на пары, так что в каждой паре скобка, стоящая ближе к левому концу слова, открывающая, а вторая – закрывающая, при этом они имеют один и тот же тип, и, кроме того, слово, стоящее между парными скобками,

является правильным скобочным выражением. Например, слова  $()$ ,  $\{()\{\}\}$ ,  $(\{\}\{\{\}\})$  – правильные скобочные выражения, а слова  $\{\{\}$ ,  $\{\{\}\}$ ,  $\{\{\}\{\}\}$  – неправильные скобочные выражения.)

1.2. «Записная книжка I». Написать программу, которая реализует функциональность телефонной записной книжки. А именно, из стандартного входа программа получает последовательность команд на добавление (*INSERT*) или поиск (*FIND*) записей. Примеры команд: *INSERT Sidorov 1234567*, *INSERT Ivanov 7654321*, *FIND Sidorov*. При выполнении команды *INSERT* программа добавляет пару (фамилия, номер) в своё хранилище и выводит строку «OK», если в хранилище нет записи с такой фамилией, или изменяет соответствующую указанной фамилии запись и выводит строку «*Changed. Old value = X*», если запись с такой фамилией уже есть в хранилище и соответствующий телефонный номер был *X*. При выполнении команды *FIND* программа выводит телефонный номер для указанной фамилии или выводит «NO», если указанной фамилии нет в справочнике. Следует использовать структуры, состоящие из двух элементов *name* и *number*. Использовать технику динамического выделения памяти для хранения записей. Оценить, как в среднем растёт число элементарных операций при выполнении команд *INSERT* и *FIND* с ростом числа хранимых записей. Записи хранить в двоичном дереве поиска.

1.3. «Записная книжка II». Решите задачу 1.2, используя хэш-таблицу с разрешением коллизий методом цепочек либо методом открытой адресации.