XXII республиканская летняя научно-исследовательская школа

**«Бригантина‑2017», 13-30 июля 2017 года**

Темы (задачи) для научных исследований по

**ИНФОРМАТИКЕ**

**Примечания.** 1) В списке тем могут дополнения и изменения.

2) Участники школы могут продолжать исследования по темам, ранее разрабатываемым в своих учебных заведениях. Для этого в первые два дня работы необходимо заявить свою тему ответственному за научные семинары по информатике и согласовать порядок работы с научным руководителем.

1. **Теория алгоритмов**
2. **Методы сортировок.**
   1. Изучить алгоритмы и написать программы, реализующие следующие методы сортировок:
      * Сортировка обменом
      * Сортировка выбором
      * Сортировка вставками (простыми и бинарными)
      * Сортировка Шелла
      * Сравнение различных длин промежутков на каждом шаге алгоритма
      * Сравнение сортировки Шелла с другими сортировками по времени работы
      * \*Исследование своего метода выбора длин промежутков на каждом шаге
      * \*Генерация «анти-тестов», т.е. тестов, которые работают максимально долго для конкретного способа выбора промежутков
      * Сортировка слиянием
      * Быстрая сортировка
      * \* Пирамидальная сортировка
      * \* Сортировка подсчётом
   2. Генерация тестов.
   3. Исследовать алгоритмы на скорость работы для массивов различной размерности.
   4. Исследовать алгоритмы на скорость работы для массивов с различной начальной расстановкой элементов (все элементы различны, возможны повторения элементов, частично упорядоченные массивы и др.).
   5. Выводы
   6. \*Создание визуализаторов, демонстрирующих различные методы сортировок
3. **Исследование связности графа.**
   1. Определение компонент связности графа
   2. Сильно связанные компоненты графа. Конденсация графа. База графа.
   3. Двусвязность. Нахождение точек сочленения и мостов графа.
   4. Использование алгоритма поиска в глубину для исследования связности (сильно связанности, двусвязнности) графа.
   5. Построение визуализаторов, для демонстрации компонент связности (двусвязности, сильно связанности) графа
4. **Пути в графе**
   1. Изучить теорию и написать программы для следующих алгоритмов поиска кратчайших путей
      * в ширину, в глубину
      * Алгоритм Флойда
      * Алгоритм Дейкстры
      * Алгоритм Форда-Бэлмана
   2. Исследовать сложность алгоритмов и количество используемой памяти для графов представленных матрицей смежности и списком смежности.
   3. \*Создание визуализаторов, демонстрирующих различные алгоритмы поиска кратчайших путей
5. **Циклы в графе**
   1. Изучить теорию и написать программы для построения циклов в графе
      * Эйлеров цикл
      * Гамильтонов цикл
      * Фундаментальное множество циклов
      * Цикл, проходящий через 2 заданные вершины
   2. Создание визуализаторов, демонстрирующих алгоритмы поиска циклов
6. **Раскраска плоского графа**
   1. Определение плоского графа. Формула Эйлера.
   2. Задача о раскраске карты.
   3. Определение минимального количества цветов, необходимого для такой раскраски объектов, чтобы не было смежных одного цвета:
      * вершины
      * рёбра
      * замкнутые области.
   4. Создание визуализаторов, демонстрирующих решение.
7. **Методы перебора**

На примере задачи о расстановке ферзей рассмотреть методы организации перебора:

* 1. полный
  2. с отсечениями
  3. с возвратом (метод ветвей и границ)

Оценить сложность алгоритмов. Написать программы, реализующие алгоритмы. Сравнить скорость работы программ.

\* Рассмотреть расстановки других фигур (ладья, слон, конь, …). Рассмотреть расстановки на нестандартных досках (частично непроходимых, цилиндрических, трёхмерных, …)

1. **Исследование методов сжатия информации** 
   1. Сжатие способом кодирования серий (RLE)
   2. Алгоритм Хаффмана
   3. Алгоритмы арифметического кодирования
2. **Современные алгоритмы шифрования**
   1. Симметричное шифрование
   2. Асимметричное шифрование
   3. Алгоритм RSA
3. **Сравнение сбалансированных деревьев поиска**
   1. Красно-чёрные деревья
   2. АВЛ-деревья
   3. \*Splay-деревья
   4. \*Декартовы деревья
   5. Другие сбалансированные деревья
4. **Конечные автоматы**
   1. изучить теорию
   2. использование конечных автоматов для синтаксического анализа строки
   3. использования конечных автоматов для вычисления значения (\*упрощения, производной) арифметического выражения
   4. рекуррентные соотношения и конечные автоматы

**Параллельное программирование**

1. **Изучение и сравнение способов решения задачи «проблема спящего парикмахера»**
   1. Простейшие решения, конфликтные ситуации
   2. Решение с помощью мьютекса
   3. Решение с помощью семафоров
   4. Вариант задачи с несколькими парикмахерами
   5. Сравнение решений по различным критериям (время простоя, количество необслуженных клиентов и т.д.)

**Машинное обучение, теория игр**

1. **Распределяющая шляпа Хогвартса**
   1. использование машинного обучения для предсказания факультета в Хогвартсе. Построение Распределяющей шляпы в виде несложного классификатора на основе черт характера, которые выделяются автоматически с HarryPotter-Wiki.
   2. использование библиотек для замены на синонимы.
   3. Приветствуется базовое знание питона.
2. **Игра в 9**

Два игрока играют на поле 3x3. Ходят по очереди. Ход заключается в том, что игрок в пустую клетку пишет цифру в диапазоне от 1 до 9. Каждую цифру можно использовать ровно один раз, в итоге таблица содержит все цифры от 1 до 9. Затем первый игрок считает суммы произведений цифр по горизонтали, а второй – по вертикали. Выигрывает игрок, набравший наибольшую сумму.

* 1. Теория игр.
  2. Дерево решений.
  3. Минимакс
  4. Отсечения
  5. Эвристический подход
  6. \*Искусственные нейронные сети.
  7. Написать программу, позволяющую проанализировать эффективность стратегий.

**Исследовательские проекты**

1. **Генерация лабиринтов**

Проанализировать представление лабиринта в памяти компьютера.

Проанализировать возможных путников.

Создать программу, создающую случайные лабиринты с заданными свойствами:

* 1. Отсутствие или наличие (возможно, ограниченного количества) циклов
  2. Отсутствие или наличие тупиков
  3. Максимальный (минимальный) путь от входа до выхода
  4. Наличие пути от входа до выхода «без левых (правых) поворотов»
  5. Наличие определённого количества «комнат» заданного размера
  6. Отсутствие прямых участков длиннее k клеток.
  7. \* Обеспечение возможности одновременного прохода двух путников (первый идёт от входа к выходу, второй обратно), если они не могут занимать одну клетку одновременно и не могут поменяться местами, если стоят на соседних клетках

1. **Исполнитель Робот**

Создать исполнителя алгоритмов, выполняющего заложенные пользователем инструкции. Приложение должно содержать поле заданного размера (не менее 1 и не превышающего 20 по вертикали или горизонтали). Должна существовать возможность редактировать, сохранять/загружать поле. Исполнитель должен уметь исполнять линейные, условные (с поддержкой операций И, ИЛИ, НЕ) и циклические операторы, а также работать в режиме прямого исполнения. Исполнитель должен выполнять команды перемещения/поворота «вперёд», «назад», «поворот направо», «поворот налево», «разворот»; проверку условий «спереди свободно». При попытке робота зайти на занятую клетку или покинуть пределы поля должно генерироваться сообщение об ошибке. Предусмотреть поддержку работы с числовыми переменными и массивами.

\* Обеспечить параллельную работу и взаимодействие нескольких исполнителей.

1. **Деловая игра Менеджмент**

Реализовать серверную и клиентскую часть игры из книги Уэзерела «Этюды для программистов». Реализовать сетевое приложение. По возможности предусмотреть создание интернет-приложения и использование алгоритмов компьютерных игроков.

\* Исследовать выигрышные стратегии.

1. **Логистика**

Есть склады с товарами и грузовики. Для каждого товара известен код, на каком складе он находится сейчас, куда его надо переместить, вес контейнера и количество контейнеров. Для каждого грузовика известен код, время проезда от каждого склада до каждого, грузоподъемность и максимальное количество контейнеров, которое он может перевозить. Необходимо реализовать график движения имеющихся грузовиков для перевозки контейнеров за минимальное время (считается время, когда последний из товаров окажется на своём месте).

\* Рассмотреть возможность с целью минимизации времени замены пунктов назначения для одноимённых товаров и использования промежуточных складов.

1. **Задачи проекта Эйлера (математические задачи, для решения которых нужно программировать)**

**ВНИМАНИЕ.** Для этих задач требуется найти не только решение, но и попытаться решить задачу в общем виде.

<http://euler.jakumo.org>

**Задача 14**

Следующая повторяющаяся последовательность определена для множества натуральных чисел:

*n* → *n*/2 (*n* - чётное)  
*n* → 3*n* + 1 (*n* - нечётное)

Используя описанное выше правило и начиная с 13, сгенерируется следующая последовательность:

13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

Получившаяся последовательность (начиная с 13 и заканчивая 1) содержит 10 элементов. Хотя это до сих пор и не доказано (проблема Коллатца (Collatz)), предполагается, что все сгенерированные таким образом последовательности оканчиваются 1.

Какой начальный элемент меньше миллиона генерирует самую длинную последовательность?

**Примечание:** После начала последовательности элементы могут быть больше миллиона.

**Задача 26**

Единичная дробь имеет 1 в числителе. Десятичные представления единичных дробей со знаменателями от 2 до 10 даны ниже:

1/2 = 0.5

1/3 = 0.(3)

1/4 = 0.25

1/5 = 0.2

1/6 = 0.1(6)

1/7 = 0.(142857)

1/8 = 0.125

1/9 = 0.(1)

1/10 = 0.1

Где 0.1(6) значит 0.166666..., и имеет повторяющуюся последовательность из одной цифры. Видно, что 1/7 имеет повторяющуюся последовательность из 6 цифр.

Найдите значение d < 1000, для которого 1/d в десятичном виде содержит самую длинную повторяющуюся последовательность цифр.

**Задача 93**

Используя каждую из цифр множества {1, 2, 3, 4} только один раз, с помощью четырех арифметических действий (+, −, \*, /) и скобок можно получить различные положительные целые числа.

К примеру,

8 = (4 \* (1 + 3)) / 2

14 = 4 \* (3 + 1 / 2)

19 = 4 \* (2 + 3) − 1

36 = 3 \* 4 \* (2 + 1)

Обратите внимание, что объединять цифры, вроде 12 + 34, не разрешается.

Используя множество {1, 2, 3, 4}, можно получить тридцать одно отличное число, среди которых наибольшим является 36. Помимо этого, до обнаружения первого числа, которое нельзя выразить данным способом, были получены все числа от 1 до 28.

Найдите множество четырех отличных цифр a < b < c < d, с помощью которых можно получить максимально длинное множество последовательных натуральных чисел от 1 до n. Ответ дайте объединив числа в строку: abcd.

**Задача 114**

В ряд в семь единиц длиной помещены красные блоки длиной как минимум три единицы так, что любые два красных блока (которые могут быть и разной длины) разделены между собой хотя бы одним черным квадратом. Существует ровно семнадцать способов этого достичь.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | spacer | spacer | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | spacer | |
| |  |  |  | | --- | --- | --- | | spacer | spacer | spacer | | |  |  |  |  | | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | | |  |  |  |  | | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | |
| |  |  |  |  | | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | | |  |  |  |  | | --- | --- | --- | --- | | spacer | spacer | spacer | spacer | | |  |  |  | | --- | --- | --- | | spacer | spacer | spacer | |
| |  |  |  | | --- | --- | --- | | spacer | spacer | spacer | | |  |  |  | | --- | --- | --- | | spacer | spacer | spacer | | |  |  | | --- | --- | | spacer | spacer | |
| |  |  | | --- | --- | | spacer | spacer | | |  | | --- | | spacer | |  |

Сколькими способами может быть заполнен ряд длиной в пятьдесят единиц?

ПРИМЕЧАНИЕ: Несмотря на то, что в примере выше это невозможно проиллюстрировать, в общем случае разрешено использовать блоки разной длины. Например, ряд длиной в восемь единиц может быть заполнен так: красный (3), черный (1), красный (4).

**Задача 148**

Можно легко показать, что ни один из элементов в первых семи рядах треугольника Паскаля не делится на 7:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | 1 |
|  |  |  |  |  | 1 |  | 1 |
|  |  |  |  | 1 |  | 2 |  | 1 |
|  |  |  | 1 |  | 3 |  | 3 |  | 1 |
|  |  | 1 |  | 4 |  | 6 |  | 4 |  | 1 |
|  | 1 |  | 5 |  | 10 |  | 10 |  | 5 |  | 1 |
| 1 |  | 6 |  | 15 |  | 20 |  | 15 |  | 6 |  | 1 |

Однако, если мы проверим первые сто рядов, то обнаружим, что только 2361 из 5050 элементов не делятся на 7.

Найдите количество элементов, которые не делятся на 7, в первом миллиарде (109) рядов треугольника Паскаля.

**Задача 204**

Число Хэмминга - такое положительное число, которое не имеет простых сомножителей больше 5. Таким образом, несколько первых чисел Хэмминга: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15. В пределах 108 существует 1105 чисел Хэмминга.

Будем называть положительное число обобщенным числом Хэмминга типа n, если у него нет простых сомножителей, превышающих n., следовательно, числа Хэмминга являются также и обобщенными числами Хэмминга типа 5.

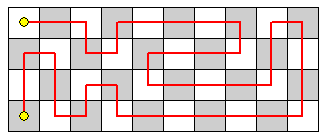
Сколько обобщенных чисел Хэмминга типа 100 можно насчитать в пределах 109?

**Задача 247**

Пусть T(n) будет количеством маршрутов по доске 4 × n таких, что:

* Маршрут начинается в левом верхнем углу.
* Маршрут состоит из движений вверх, вниз, влево или вправо на одну клетку.
* Маршрут проходит по каждой клетке ровно один раз.
* Маршрут кончается в нижнем левом углу.

Рисунок ниже показывает один из маршрутов по доске 4 × 10:

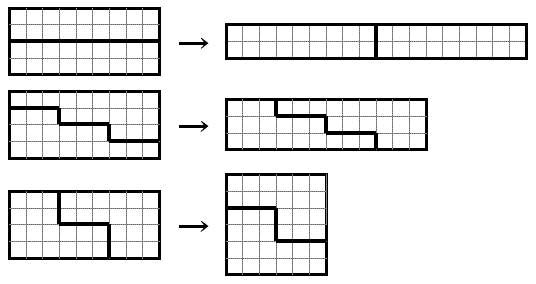


T(10) равно 2329. Чему равно T(1012) mod 108?

**Задача 338**

Возьмем прямоугольный лист бумаги в клеточку размерами w × h. Шаг клетки составляет 1. Если разрезать такой лист по границам клеток и совместить полученные куски без перекрытия, то можно образовать прямоугольники других размеров.

К примеру, из листа размерами 9 × 4, можно создать прямоугольники размерами 18 × 2, 12 × 3 и 6 × 6, если разрезать и соединить полученные куски, как это показано ниже:



Аналогично, из листа размерами 9 × 8 , можно образовать прямоугольники размерами 18 × 4 и 12 × 6 .

Для пары чисел w и h определим F(w,h) как количество отличных прямоугольников, которые можно образовать из листа бумаги размерами *w* × *h* . Например, F(2,1) = 0, F(2,2) = 1, F(9,4) = 3 и F(9,8) = 2. Учтите, что прямоугольники, конгруэнтные заданному, не учитываются в F(w,h). Также учтите, что прямоугольник размерами w × h и прямоугольник размерами h × w не считаются отличными.

Для целых значений N определим G(N) в виде суммы F (w,h) для всех тех пар w и h, которые удовлетворяют требованию 0 < h ≤ w ≤ N. Можно убедиться, что G(10) = 55, G(103) = 971745 и G(105) = 9992617687.

Найдите G(1012). Ответ приведите по модулю 108.

**Задача 463**

Странные рекурсивные отношения

Функция *f* определена для всех положительных целых аргументов следующим образом:

* *f*(1)=1
* *f*(3)=3
* *f*(2*n*)=*f*(*n*)
* *f*(4*n*+1)=2*f*(2*n*+1)−*f*(*n*)
* *f*(4*n*+3)=3*f*(2*n*+1)−2*f*(*n*)

Функция *S*(*n*) определена как . *S*(8)=22 и *S*(100)=3604.

Найдите *S*(337). В качестве ответа приведите последние 9 цифр полученного числа.

**Задача 466**

Различные элементы таблицы умножения

Пусть P(*m*,*n*) будет количеством *различных* элементов в таблице умножения *m*×*n*.

Например, таблица умножения 3×4 выглядит следующим образом:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **×** | **1** | **2** | **3** | **4** |
| **1** | 1 | 2 | 3 | 4 |
| **2** | 2 | 4 | 6 | 8 |
| **3** | 3 | 6 | 9 | 12 |

В ней 8 различных элементов {1,2,3,4,6,8,9,12}, поэтому P(3,4) = 8.

Вам дано, что:

P(64,64) = 1263,

P(12,345) = 1998 и

P(32,1015) = 13826382602124302.

Найдите P(64,1016).

**Задача 477.**

Игра в числовые последовательности

Игра в числовые последовательности начинается с последовательности *S* из *N* чисел, записанных в ряд.

Два игрока ходят по очереди. В свой ход каждый игрок должен выбрать или первое, или последнее число в последовательности и убрать его из нее. Очки игрока равны сумме всех чисел, которые были им убраны. Каждый игрок стремится получить как можно большее количество очков.

Если *N* = 4 и *S* = {1, 2, 10, 3}, тогда каждый игрок получает максимальную возможную сумму следующим образом:

* Игрок 1: убирает первое число (1)
* Игрок 2: убирает последнее число из оставшейся последовательности (3)
* Игрок 1: убирает последнее число из оставшейся последовательности (10)
* Игрок 2: убирает оставшееся число (2)

Игрок 1 получает 1 + 10 = 11 очков.

Пусть *F*(*N*) будет количеством очков игрока 1, если оба игрока следуют оптимальной стратегии игры для последовательности *S* = {*s*1, *s*2, ..., *sN*}, определенной как:

* *s*1 = 0
* *si*+1 = (*si*2 + 45) modulo 1 000 000 007

Последовательность начинается следующим образом: *S* = {0, 45, 2070, 4284945, 753524550, 478107844, 894218625, ...}.

Вам дано, что *F*(2) = 45, *F*(4) = 4284990, *F*(100) = 26365463243, *F*(104) = 2495838522951.

Найдите *F*(108).

1. **Разработка проектов**

**(для младшей группы)**

1. **Калькулятор.** Разработать простейший калькулятор (возможно, с длинной арифметикой).

**\*** Можно ли получить любое целое неотрицательное число (не более 9 разрядов), если часть кнопок-цифр работать не будет, но можно делать арифметические операции?

1. **Блокнот.** Разработать программу, аналогичную Windows-приложению Блокнот (возможно, с элементами подсветки зарезервированных слов).

**\*** Как можно быстро определить, находится ли набранное слово в списке зарезервированных?

1. **MyPaint.** Разработать простейший графический редактор (возможно, с такими дополнительными инструментами, например, зеркало, двойное зеркало, калейдоскоп, …).

\* Чем будет отличаться создание векторного редактора от растрового?

1. **Часы.** Разработать программу, которая может отображать часы в электронном виде или в виде со стрелками. Добавить функции таймера, секундомера, будильника, календаря, органайзера и т.п.

**\*** Какие календари существовали?Что такое високосный год?Какопределить день недели по дате?

1. **Движение.** Разработать программу, с помощью которой можно управлять движением картинок по начальному и конечному положению картинок, осуществлять выбор траекторий движения и т.д.

**\*** Какие системы координат Вы знаете? Что такое вектор? Что такое спрайт? Какие механизмы реализации графики вы знаете?

1. **Угадай число.** Разработать программу, реализующую роль ведущего в игре об угадывании числа от 1 до 100 не более чем за 10 вопросов. Вопросом является число. Ведущий отвечает, задуманное число больше, меньше, либо совпадает с числом-вопросом.

**\*** За какое минимальное число вопросов можно найти загаданное число в худшем случае?

1. **Крестики-нолики.** Разработать программу, позволяющую играть в крестики-нолики между двумя игроками, или игроку и компьютеру (случайные ходы или выигрышная стратегия).

**\*** Как описать выигрышную стратегию?А в игре «пять в ряд»?