Задачи очного тура олимпиады
по дискретной математике и теоретической информатике (ДМиТИ-2015)

Описываемые ниже задачи основаны на тех же понятиях и манипуляторах, что и задачи заочного тура. Поэтому участники уже знакомы с такими понятиями как связность графа, планарность, равенство (изоморфность) графов, регулярное выражение, машина Тьюринга, кванторы и т. д.
Участникам предлагается сначала решать задачи без «звёздочки». За каждую по 3 балла. Затем задачи со «звёздочками», которые оцениваются по 6 баллов и рассматриваются как дополнительные, то есть участник может решать те, которые ему больше понравятся.
Время решения 3 астрономических часа.

1. *«Графы»*. *К задаче прилагается манипулятор, в котором можно нарисовать граф и, перемещая вершины, исследовать его на планарность (манипулятор устроен так, что при перемещении вершин, они не совмещаются). Наличие манипулятора позволяет избежать формальных определений связности и планарности (хотя при нажатии кнопки «помощь» они выводятся).*

1.1. В окне манипулятора представлен связный граф с 7 вершинами, у которого в каждой вершине сходятся два ребра (степень вершины равна 2) и рёбра не пересекаются. Добавьте в граф как можно больше новых рёбер так, чтобы степени всех вершин оставались одинаковыми, а граф - планарным (то есть, его вершины можно было переместить так, чтобы рёбра не пересекались).
1.2*. Пусть *f(n)* - максимальная степень вершины в связном *регулярном* планарном графе с *n* вершинами (регулярность означает, что степени всех вершин равны). Приведите и обоснуйте оценки сверху и снизу для *f(n).*

2. Комбинаторика. *К задаче прилагается манипулятор, позволяющий строить графы. Понятие изоморфности графов формулируется в «терминах манипулятора» и легко проверяется практически.*

2.1. Постройте все различные (неизоморфные - которые нельзя преобразовать друг в друга перетаскиванием вершин) связные графы, имеющие 6 вершин и ровно один цикл длины 3 (состоящий из 3 рёбер), (один из таких графов уже представлен в манипуляторе).
2.2*. Сколько существует различных (неизоморфных) связных графов, имеющих *n* вершин и ровно один цикл, включающий *n*-3 вершины?
<u>Примечание</u>. Достаточно рассмотреть только значения n, при которых n-3 кратно 3. Также правильными будут считаться ответы, в которых дан не точный ответ на поставленный вопрос, а оценки сверху и снизу.

3. Регулярные выражения+комбинаторика. *К задаче прилагается манипулятор, проверяющий принадлежность объектов множеству, описанному регулярным выражением.*

3.1. Постройте регулярное выражение, которое описывает все правильные в математическом смысле формулы, в которые входят только числа 2, операции + и * и квадратные скобки. В этих формулах запрещено помещать квадратные скобки внутрь других квадратных. Также в скобках не может стоять одна буква без знака операции. Пустой символ обозначается Λ (добавление пустого символа в любом месте не меняет строки). Пустая строка считается правильным выражением.
Примеры правильных выражений: 2; 2+2*2; [2+2+2]*2+[2+2]; 2+2*[2+2]+2
Примеры неправильных выражений: [2]; [2*[2+2]]
(^ - символ итерации, | - символ альтернативности выбора)

3.2. Сколько формул, подчиняющихся этим правилам, содержат ровно 11 символов (здесь учитываются все формы записи, даже если математически результат у некоторых формул будет одинаковым)?

4. Логические схемы. *Задача построена на основе манипулятора, в котором можно собирать схемы из элементов И, ИЛИ, НЕ (каждый сигнал можно подавать на входы нескольких элементов).*
4.1. Сконструировать логическую схему, на входы x, y которой подаётся двоичное число <u>ab</u>, а на выходах c, d, e число <u>cde</u>, на единицу большее <u>ab</u>.
4.2*. Оцените сверху число элементов схемы, реализующей сложение с единицей двоичного числа с *n* цифрами.

5. Логика высказываний и предикатов. *Работа идёт с манипулятором «Мир Тарского». В этом манипуляторе можно делать и проверять на примерах утверждения относительного фигур, стоящих на клетчатой прямоугольной доске, на которых определены одноместные предикаты: «красный», «синий», «большой», «малый», «шар», «куб»; двухместные предикаты «рядом с», «левее» (в столбце, расположенном левее), «выше» (в строке, расположенной выше).*

5.1. Запишите логическое выражение, описывающее те и только те конфигурации, в которых все фигуры стоят подряд на одной горизонтали, причём крайние фигуры имеют синий цвет, а внутренние — красный.
5.2*. Постарайтесь по возможности уменьшить число кванторов в построенном логическом выражении и объяснить эквивалентность преобразованных выражений исходным.

6. Математическое определение алгоритма: машина Тьюринга. *Работа идёт с манипулятором, эмулирующим машину Тьюринга над алфавитом {0; 1; a; b; *}. Где символ * обозначает «пустой» символ.*
6.1. Машина Тьюринга T1 копирует цепочку нулей и единиц, идущих подряд, располагая их правее заданной цепочки и разделяя цепочки символом * (которым исходно заполнены все свободные ячейки ленты). Головка машины в начале и в конце выполнения алгоритма указывает на первую слева единицу. Например, цепочка <u>1</u>11011 преобразуется в цепочку <u>1</u>11011*111011.
Машина T2 вычитает из числа в единичной системе, стоящего первым, следующее за ним и меньшее его число (символы, стоящие левее вычитаемого, сохраняются). Головка машины в начале выполнения алгоритма указывает на первую слева единицу вычитаемого, которая не меняет своего положения на ленте. В конце головка указывает на первую слева единицу результата. Например, цепочка 1*<u>1</u>1111*11 преобразуется в цепочку 1*<u>1</u>11.
Машина T3 перемножает числа в единичной системе счисления, например, цепочка <u>1</u>11*11 преобразуется в цепочку <u>1</u>11111 (все остальные ячейки машины, кроме указанных цепочек, заполнены символами *).
В работе вышеперечисленных машин используются вспомогательные символы 0, a и b.
Комбинируя эти машины, требуется построить машину Тьюринга T, которая выполняет над двумя числами в единичной системе счисления операцию A^2-B^2 (A>B).
Например, цепочка <u>1</u>11*11 должна преобразоваться в цепочку <u>1</u>1111.
<u>Примечание*</u>. Все состояния машин T1, T2 и T3, кроме начального и конечного, различны.

6.2*. Сколько ячеек будет использовано (использованные ячейки это те, в которые машина записывает какой-либо символ по крайней мере один раз) построенной машиной Тьюринга T, если число A состоит из n единиц, а число B из m единиц? Правильным ответом будут считаться и оценки сверху и снизу для искомого числа.

* Приложение. Машины Т1, Т2, Т3 с комментариями:

Машина Т1 – копирование строки из 0 и 1

s0 [1] -> p1 [a] R
обозначение копируемой 1 за «a»
p1 [1] -> p1 [1] R
p1 [0] -> p1 [0] R
p1 [*] -> p2 [*] R
переход вправо на место копирования строки
p2 [0] -> p2 [0] R
p2 [1] -> p2 [1] R
p2 [*] -> p3 [1] L
копирование 1 на первое свободное место
p3 [0] -> p3 [0] L
p3 [1] -> p3 [1] L
p3 [*] -> p3 [*] L
p3 [a] -> p4 [a] R
переход к следующему копируемому символу
p4 [1] -> p1 [a] R
p4 [0] -> p5 [b] R
s0 [0] -> p5 [b] R
копирование 0
p5 [0] -> p5 [0] R
p5 [1] -> p5 [1] R
p5 [*] -> p6 [*] R
p6 [0] -> p6 [0] R
p6 [1] -> p6 [1] R
p6 [*] -> p3 [0] L
размещение копии 0
p3 [b] -> p4 [b] R
p4 [*] -> p7 [*] L
p7 [a] -> p7 [1] L
p7 [b] -> p7 [0] L
замена символов «a» и «b» на 1 и 0 соответственно
p7 [*] -> f [*] R

Машина Т2 – вычитание чисел в единичной системе счисления

s0 [1] -> s0 [1] R
s0 [*] -> t1 [*] R
поиск начала второго числа
t1 [a] -> t1 [a] R
t1 [1] -> t2 [a] L
обозначение вычтенных единиц символами «a»
t2 [a] -> t2 [a] L
t2 [*] -> t2 [*] L
t2 [1] -> t3 [*] R
поиск и вычитание очередной единицы из вычитаемого
t3 [*] -> t3 [*] R
t3 [a] -> t1 [a] N
зацикливание процесса

t1 [*] -> t4 [*] L
t4 [a] -> t4 [*] L
уничтожение символов «а» после окончания вчитаия
t4 [*] -> t4 [*] L
t4 [1] -> t5 [1] L
t5 [1] -> t5 [1] L
t5 [*] -> f [*] R
установка считывающей головки на начало

Машина Т3 – умножение чисел в единичной системе счисления

s0 [1] -> v1 [*] R
состояние 1 запоминает, что вторoе число надо скопировать в результат
первое число используется как счетчик сложения второго числа
v1 [1] -> v1 [1] R
пропуск оставшихся единиц первого множителя
v1 [*] -> v2 [*] R
состояние 2 символизирует переход ко второму множителю
v2 [1] -> v2 [a] R
замена символами «а» всех единиц, подлежащих копированию
v2 [*] -> v3 [*] L
состояние 3 символизирует начало копирования
v3 [1] -> v3 [1] L
поиск первого нескопированного символа второго множителя
v3 [a] -> v4 [1] R
начало копирования
v4 [1] -> v4 [1] R
пропуск уже скопированных единиц второго множителя
v4 [*] -> v5 [*] R
переход к результирующей части записи
v5 [1] -> v5 [1] R
поиск конца промежуточного результата
v5 [*] -> v6 [1] L
добавление очередной скопированной единицы
v6 [1] -> v6 [1] L
возвращение ко второму множителю
v6 [*] -> v3 [*] L
повторение копирования единиц второго множителя
v3 [*] -> v7 [*] L
v7 [1] -> v7 [1] L
переход к первому множителю (счётчику) после окончания очередного копирования второго множителя
v7 [*] -> s0 [*] R
повторение цикла копирования ля очередной единицы первого слагаемого
s0 [*] -> v8 [*] R
v8 [1] -> v8 [*] R
стирание единиц второго множителя
v8 [*] -> f [*] R
установка считывающей головки на начало результата