Часто задаваемые вопросы

Можно выполнять все задания из списка заранее и получить баллы?
Да. Когда задачу разберем на семинаре – она потеряет в цене.

Задания "сформулировать ..." и анализом сложности сдаются на семинарах?
Можно сдать листочек с разбором на семинаре. Можно прислать по почте .doc/.pdf файл или фотографию листочка.

Программы присылать, подставляя конкретные параметры, где нужно?
Для задач с параметрами, например "сложить 2 числа в k-ричной системе", надо:

- написать одну программу для конкретного k (не такого, какой разбирали на семинарах),
- описать общий алгоритм – как генерировать программу для произвольного k.

Как включить третью ленту для МТ? Ну, и, для будущего – 4 или больше.
Три бесконечные вправо ленты:

```
tapes = [right, right, right]
```

"Записанное наоборот" число из задач к первому семинару — это как если бы лента уходила влево и мы писали число у её конца?
Число 12 (в десятичной) будет записано на ленте вот так: #0011#.

Возможно ли в эмуляторе задать правило $\# \to aRR$ (несколько шагов)?
Нет, придется записывать как несколько правил.

1 1-ленточные МТ

В задачах данного раздела предполагается, что у МТ одна бесконечная вправо лента.

Задача 1.1. Написать программу прибавления 1 к двоичному числу, записанному "наоборот". Провести анализ в среднем/в худшем. (Разбиралось на семинарах, 0 баллов).

Задача 1.2. Написать программу прибавления 1 к двоичному числу, записанному "правильно". Провести анализ в среднем/в худшем. (1 балл).

Задача 1.3. Написать программу прибавления 3 к двоичному числу, записанному "наоборот". (Разбиралось на семинарах, 0 баллов). Провести анализ в среднем/в худшем (в зависимости от простоты получившейся формулы для сложности в среднем, 1-3 балла).

Задача 1.4. Сформулировать, как пишется программа прибавления k к двоичному числу, записанному "наоборот". (Разбиралось на семинарах, 0 баллов). Провести анализ в среднем/в худшем (в зависимости от простоты получившейся формулы для сложности в среднем, 1-3 балла).

Задача 1.5. Сформулировать, как пишется программа прибавления k к m-ричному числу, записанному "наоборот". (1 балл).

Задача 1.6. Написать программу разворота бинарной строки в алфавите $\{\#, 0, 1\}$ (1 балл). Примеры:

```
#       -> #
#0#     -> #0#
#01#    -> #10#
#11001# -> #10011#
```

Провести анализ в среднем/в худшем (2 балла).

Задача 1.7. Алфавит $=\{\#, a, b\}$. Написать программу, которая находит самое левое вхождение подстроки "aababab"во входной строке и стирает с ленты все символы, ДО этого вхождения. Если искомая подстрока не нашлась – лента должна быть пуста. (1 балл)

```
#             -> #
#ab#          -> #
#aababab#     -> #aababab#
#abababaababab#-> #####aababab#
```

Доказать оптимальность в худшем случае (10 баллов.)

В задачах 1.8 - 1.9 предполагается, что если входные данные являются строкой из распознаваемого языка, то лента должна опустеть. Иначе – программа должна перейти в ошибочное состояние.

Задача 1.8. Распознаватель $a^n b^n$ (0 баллов). См. samples. Анализ в худшем/среднем (1 балл)

Задача 1.9. Распознаватель $a^n b^n c^n$ (1 балл). Анализ в худшем/среднем (2 балла)

Задача 1.10. Алфавит $= \{\#, a, b\}$. Зафиксирована строка db = "aababababaabbbababaaab". Написать программу, которая проверяет, что строка, записанная на ленте, является подстрокой db. Если это так – в конце работы программы лента должна стать пустой, иначе – программа должна перейти в ошибочное состояние (1 балл). Если доказано, что решение оптимально – еще 1 балл.

Примеры:

```
# -> #
#a# -> #
#aaaa# -> Error
#abababaa# -> #
```

Рассчитать сложность в среднем (3 балла).

2 Многоленточные МТ

В задачах данного раздела предполагается, что у МТ несколько лент с одинаковым алфавитом.

Задача 2.1. Написать программу перевода числа, записанного в двоичной системе, в восьмеричную. Считается, что число записано на первой ленте "наоборот", результат должен быть написан на второй ленте "наоборот", других изменений с лентами не должно произойти. Пример:

```
--------------------------
    #         -> #
    #         -> #
--------------------------
    #0#       -> #0#
    #         -> #0#
--------------------------
    #0001111# -> #0001111#
    #         -> #071#
```

Доказать оптимальность в худшем случае (2 балла).

Задача 2.2. Написать программу перевода числа, записанного в троичной системе, в 11-ричную. Считается, что число записано на первой ленте "наоборот", результат должен быть написан на третьей ленте "наоборот". Вторая лента для промежуточных вычислений. Перед завершением работы, программа должна очистить вторую ленту (заполнить символом #). (2 балла).

Задача 2.3. Сложить 2 числа, записанных в k-ричной системе счисления и записать результат на третью ленту (1 балл). Выбрать $k \in \{5, 6, 7\}$.

Задача 2.4. Умножить 2 числа в k-ричной системе (1 балл).

Задача 2.5. Поделить нацело 2 числа в k-ричной системе (1 балл).

Задача 2.6. Поделить с остатком 2 числа в k-ричной системе (1 балл).

Задача 2.7. Алфавит = $\{a, b, c\}$, 2 ленты. На первой ленте записано слово $v = \#(a|b|c)^*\#$, на второй $w = \#(a|b|c)^*\#$. Проверить, что v является подстрокой w (если ответ – "да", то ленты должны остаться неизменными, иначе – ошибка) (1 балл). Провести анализ в среднем (2 балла).

3 RAM

3.1 Алгоритмы

В каждой задаче данного раздела необходимо оценить сложность в худшем при равномерном весовом критерии.

Задача 3.1.1. Вывести последовательность чисел от l до r с шагом s, если $r \geq l$ (0.5 балла):

```
in : l, r, s
out: l, l+s, ..., r
```

Задача 3.1.2.. Вывести числа в последовательности $a1, ..., an$, которые делятся без остатка на d (0.5 балла):

```
in : d, n, a1, ..., an
out: b1, ..., bk
```

Задача 3.1.3.. Определить, является ли последовательность $a1, ..., an$ палиндромом. Вывести 1, если является, или 0, если не является (1 балл):

```
in : n, a1, ..., an
out: r in {0,1}
```

Задача 3.1.4.. Вывести двоичное представление десятичного числа.

- least significant bit справа – 0.5
- LSB слева – 0.5

```
in : x  (1 регистр)
out: y  (log2(x) регистров)
```

Задача 3.1.5.. Определить, является ли последовательность $a1, ..., an$ отсортированной в порядке возрастания элементов, то есть $a1 \leq ... \leq an$. Вывести 1, если является, или 0, если не является (0.5 балла):

```
in : n, a1, ..., an
out: r in {0,1}
```

Задача 3.1.6.. Отсортировать последовательность $a1, ..., an$ в порядке возрастания элементов:

- Пузырек - 0.5
- QuickSort - 1.5
- MergeSort - 1.5
- Любая другая сортировка
  - 1.0, если доказана сложность в среднем как $O(nlogn)$,
  - 0.5, , если доказана сложность в среднем как $O(n^2)$.
- Известно, что $|ai| \leq 100$. Реализовать сортировку подсчетом (сложность $O(n)$ при равномерном весовом критерии). 2 балла.

```
in : n, a1, ... , an
out: b1, ... , bn
```

Задача 3.1.7.. Алгоритм быстрого возведения в степень.

```
in : n
out: n^n
```

- 0.5 - за работающее решение
- 1.0 - за решение, использующее $O(log_2(n))$ умножений

Задача 3.1.8.. Проверить, является ли входное число $n \geq 2$ простым (имеет ровно 2 делителя – 1 и n). Вывести 1, если является, или 0, если не является.

```
in : n
out: r in {0,1}
```

0.5 балла за "наивную"реализацию. Использование нетривиальных алгоритмов поощряется дополнительными баллами.

3.2 Program transformation and analysis

Задача 3.2.1. Text relocation (2 балла).
Вход: $shift \geq 1, n \geq 0, A[0, 1, ..., 2n-2, 2n-1]$, где A – корректная программа для RAM (каждая команда кодируется 2 регистрами – опкод команды + аргумент).
Выход: программа для RAM следующего вида:

```
JUMP shift+1
Halt 0
...
Halt 0  (shift-1 раз)
B[0]
...
B[2n-1]
```

Подразумевается, что B – программа для RAM, отличающаяся от A тем, что аргументы инструкций JUMP, JZERO и JGTZ увеличены на shift. Таким образом, результатом работы алгоритма является программа, полностью эквивалентная исходной (с той же словарной функцией).
Задача 3.2.2. Написать программу, которая печатает свой собственный текст (5 баллов).

4 RASP

Задача 4.1. Загрузчик RASP программ (5 баллов).

Вход: $n, A[0...2n-1], arg_1, ..., arg_n$, где A – корректная программа для RASP (каждая команда кодируется 2 регистрами – опкод команды + аргумент).

Необходимо загрузить A в регистры $R_1, ..., R_{2n}$, занулить R_0, передать управление на R_1.

Замечание: $arg_1...arg_n$ считывать с входной ленты НЕЛЬЗЯ.

Таким образом, результатом работы алгоритма является эмуляция входной программы на заданных аргументах.

Задача 4.2. Написать программу, которая печатает свой собственный текст (2 балла).