

## Домашнее задание 1.

### 1.

Постройте машину Тьюринга, вычисляющую функцию  $n \mapsto n \bmod 5$  в унарной кодировке аргументов. Оцените время вычислений на вашей машине в зависимости от  $n$ .

#### Описание:

Машина будет проходиться по слову-числу, удаляя единицы, и при этом циклично меняя свое состояние среди диапазона  $\{q_1, \dots, q_5\}$ . После удаления всех единичек, остаток от деления на 5 будет соответствовать номеру состояния, и останется лишь записать номер этого состояния.

#### Построение:

- $q_s, \# \rightarrow q_1, \#, R \parallel$  Старт
- $q_1, 1 \rightarrow q_2, \#, R \parallel$  Проходим по числу, циклично меняя состояния
- $q_2, 1 \rightarrow q_3, \#, R$
- $q_3, 1 \rightarrow q_4, \#, R$
- $q_4, 1 \rightarrow q_5, \#, R$
- $q_5, 1 \rightarrow q_1, \#, R$
- $q_5, \# \rightarrow q_4, 1, R \parallel$  Дойдя до конца, начинаем записывать остаток, пока не вернемся на состояние  $q_1$
- $q_4, \# \rightarrow q_3, 1, R$
- $q_3, \# \rightarrow q_2, 1, R$
- $q_2, \# \rightarrow q_1, 1, R$
- $q_1, \# \rightarrow q_b, 1, 0 \parallel q_b$  – состояние для возвращения назад после написания остатка
- $q_b, 1 \rightarrow q_b, 1, L$
- $q_b, \# \rightarrow q_f, \#, 0$

### 2.

Опишите машину Тьюринга, вычисляющую функцию  $n, m \mapsto n * m$  в унарной кодировке аргументов. Аргументы разделяются специальным символом  $\$ \in \Gamma \setminus \Sigma$ . Оцените время вычислений на вашей машине в зависимости от длины входа.

#### Описание:

Начальная конфигурация:  $1^n 1^m$ .

1. Машина заменяет первую единицу в левом числе на  $\#$  и переходит в состояние копирования числа  $m$ .
2. Машина копирует число  $m$  и приписывает результат его справа.
3. Цикл повторяется, пока от левого числа ничего не останется.
4. Последним шагом машина убирает все лишнее.

Сложность: Копирование числа  $m$ :  $O(m * (m * n))$  (поскольку записываем скопированное число справа от результата). Всего  $n$  таких копирований. Итого  $O(m^2 n^2)$ .

### 3.

Опишите машину Тьюринга, за время  $O(n \log n)$ :

- а) преобразующую унарную запись числа  $n$  в бинарную;
- б) преобразующую бинарную запись числа  $n$  в унарную;

- с) вычисляющую функцию  $n \mapsto 2n$  в унарной кодировке.

### Описание:

а)

1. Головка проходит по унарной записи числа, удаляя каждую вторую встретившуюся единицу —  $O(n)$ .
2. Головка записывает слева от унарной записи числа 1 или 0 в зависимости от последнего состояния на унарном числе перед встречей с # (была удалена единица на прошлом шаге или нет). Каждая последующая запись происходит слева от записанного промежуточного результата.

Итого:  $\log n$  шагов, каждый сложностью  $O(n)$ , всего  $O(n \log n)$ .

### Пример:

1111111  $\rightarrow$  111

1. ###1111111#  $\rightarrow$  ###1010101#  $\rightarrow$  #1#1010101#
2. #1#1010101#  $\rightarrow$  #1#1001000#  $\rightarrow$  #11#1001000#
3. #11#100100#  $\rightarrow$  #11#1000000#  $\rightarrow$  111#1000000#

б)

1. Вычитается единица из бинарного числа —  $O(\log n)$ .
2. Ставится единица перед бинарным числом —  $O(1)$ , так как головка уже сдвинулась к началу числа.
3. Бинарная запись числа сдвигается на 1 вправо —  $O(\log n)$ .

Итого:  $n$  шагов, каждый сложностью  $O(\log n)$ , всего  $O(n \log n)$ .

с)

1. Конвертация в бинарную запись —  $O(n \log n)$ .
2. Удвоение путем приписывания 0 справа в бинарной записи —  $O(1)$ .
3. Конвертация удвоенного числа в унарную запись —  $O(n \log n)$ .