Contents

1 Problem 2.4 **2**

2 Problem 2.5a **2**

3 Problem 2.5b **2**

4 Problem 2.5c **3**

5 Problem 2.5d **3**

6 Problem 2.6 **4**

7 Problem 2.7 **4**

8 Problem 2.9 **5**

9 Problem 2.11 **6**

1 Problem 2.4

Нехай стандартна багатострічкова МТ виконує задачу за час t.

Порівняємо виконання тієї ж задачі на стандартній багатострічковій МТ (МТ1) та на довільній багатоканальній МТ (МТ2), на якій зчитувальні пристрої рухаються одночасно:

> Якщо на МТ1 зчитувальний пристрій на якійсь стрічці i рухається вправо, а на іншій $j : j \neq i$ рухається вліво, то для МТ2 ці дії на обох стрічках на будуть проводитися паралельно. Натомість потрібно буде закінчити дію на стрічці i, не змінюючи стрічку j, та повернутися до зміни стрічки j, не змінюючи стрічку i.

Тобто для певної дії ζ, яка займає k_ζ тактів на МТ1 на МТ2 вона займе $k_\zeta + r_\zeta + k_\zeta = 2k_\zeta + r_\zeta$, де r_ζ це кількість тактів на повернення.

Таким чином для МТ1, що виконує задачу за $t = \sum k_\zeta$ тактів на МТ2 ця задача займе:

$$\sum_\zeta (2k_\zeta + r_\zeta) = 2t + R$$

2 Problem 2.5a

1. Маємо дві стрічки: вхідну та вихідну.
2. Проходимо вхідну стрічку записуючи результат проходу в вихідну:
   (a) Символ c та b у стані q_0 записуємо без змін.
   (b) На символі a переходимо у новий стан, що буде означати очікування символу b для заміни $ab \to c$ (при переході на новий описаний стан не записується символ a на вихідну стрічку)
   (c) Якщо на стані очікування зчитуємо символ b, то записуємо символ c та рухаємося далі. Інакше записуємо символ a та поточний зчитанний.

3 Problem 2.5b

1. Дві стрічки: вхідна (зчитуємо) та вихідна (записуємо).
2. Копіюємо вхідне слово на вихідну стрічку.
3. Повертаємося на вхідній стрічці на початок.
4. Знову копіюємо вхідне слово на вихідну стрічку (зчитувальний пристрій вихідної стрічки при поверненні на кроці 3 залишається на місці, тобто на першому пустому символі після слова)

4 Problem 2.5c

1. Дві стрічки: вхідна (зчитування) та вихідна (запис).
2. Записуємо нуль із вхідної на вихідну.
3. Копіюємо всі одиниці із вхідної, поки не зустрінемо нуль та записуємо його.
4. Повертаємося на вихідній стрічці на початок.
5. Копіюємо решту вхідного слова, рухаючись вліво по вихідній стрічці.

5 Problem 2.5d

1. Маємо чотири стрічки: вхідну, стрічку для першого числа, стрічку для другого числа, вихідну стрічку.
2. Копіюємо все із вхідної стрічки до символу 2 на стрічку для першого числа.
3. Копіюємо все з символу 2 до пустого символу на стрічку для другого числа.
4. Починаємо додавання двох чисел із стрічок для двох чисел на вихідну стрічку:
   (a) Переводимо зчитувальні пристрої стрічок двох чисел на кінець вправо цих чисел.
   (b) Зчитувальні пристрої на стрічках чисел рухаються одночасно вліво.
   (c) Додаємо два поточних символа за такими умовами:
      i. Якщо 1 0, 0 1, 1 #, # 1, то записуємо 1 та рухаємося далі.
      ii. Якщо 0 0, то записуємо 0 та рухаємося далі без переповнення.
      iii. Якщо 1 1, то переходимо в стан переповнення та записуємо 0.
      iv. Якщо 1 0, 0 1, 1 #, # 1 та стан переповнення, то записуємо 1, переходимо в нормальний стан
      v. Якщо 0 0 та стан переповнення, то записуємо 1 та переходимо в нормальний стан.
      vi. Якщо 1 1 та стан переповнення, то записуємо 1 та залишаємося в стані переповнення.
      vii. Якщо # # та стан переповнення, то записуємо 1 та завершуємо роботу.
      viii. Якщо # # та в не в стані переповнення, то завершуємо роботу.

6 Problem 2.6

1. n

   Побудована машина Тюрінга для доказу конструктивності за часом функції n копіює вхідне слово із одиниць 1^n з вхідної стрічки на вихідну за n тактів.

2. $nlogn$

   Побудована машина Тюрінга для доказу конструктивності за часом функції $nlogn$ перетворює вхідне слово із вхідної стрічки у двійкове на додаткову стрічку №1. Це перетворення описується так: на кожний прочитаний вхідний символ машина перезаписує двійкове число на стрічці №1. Максимальна кількість операцій для перезапису на вхідному слові 1^n буде $logn$. Оскільки цей перезапис буде проводитися після прочитання кожного вхідного символу, то можемо грубо оцінити його в $O(nlongn)$ операцій. Далі вхідне число з стрічки №1 записується на вихідну стрічку за $logn$ разів, де лічильником слугує слово на стрічці №1. Отримали таку машину Тюрінга, що перетворює $1^n \to 1^{nlogn}$.

3. n^2

   Машина Тюрінга для доказу конструктивності за часом функції n^2 описана таким чином:

   (a) Маємо дві стрічки: вхідна та вихідна.

   (b) Копіюємо слово з вхідної стрічки на вихідну знову таким чином, що після k копіювань на вихідній стрічці маємо слово 1^{kn}.

   (c) Повторюємо кроки 2 поки можемо.

   Крок 2 повторюється n разів, копіювання займає n тактів, а отже дана МТ працює за $t(n) = n^2$.

4. 2^n

   Будуємо машину Тюрінга із трьома стрічками. Записуємо символ '1' на додаткову та вихідну стрічки. Потім на кожний зчитувальний символ першої стрічки дописуємо все слово з додаткової стрічки на вихідну та попереднє слово з вихідної на додаткову. Кожне k копіювання займає $O(2^k)$.

7 Problem 2.7

Нехай існує така машина Тюрінга, яка працює $\lceil logn \rceil$ тактів та для вхідного слова 1^n повертає 1^{logn}.

Розглянемо два слова довжини n_1 та довжини $n_2 = n_1 + 1$. Слово може бути зчитано за n тактів. Але ми маємо лише $\lceil logn \rceil$ тактів. Таким чином для слів довжини

n_1 та n_2 машина Тюрінга буде зчитувати лише $\lceil logn_1 \rceil$ та $\lceil logn_2 \rceil = \lceil log(n_1 + 1) \rceil = \lceil logn_1 \rceil$, а отже видавати однаковий результат. Маємо протиріччя.

8 Problem 2.9

Нехай існують такі машини Тюрінга:

1. $$M_f : 1^n \to 1^{f(n)}$$

   Працює за $O(f(n))$. Має l_f стрічок.

2. $$M_g : 1^n \to 1^{g(n)}$$

   Працює за $O(g(n))$. Має l_g стрічок.

1. $h_1 = f(n) + g(n)$

   Будуємо машину Тюрінга M_{h_1}, що має $l_{h1} = l_f + l_g - 2$ стрічок. Спочатку працює підмашина M_f на перших l_f стрічках, що займе $O(f(n))$ тактів та результат буде записано на l_f стрічці. Потім працює підмашина M_g, у яка працює з $l_f + 1$ до $l_f + l_f - 2$ стрічки та на першій стрічці, та займає $O(g(n))$ часу. Вихідна стрічка це $l_f + l_g - 2$ стрічка. M_{h_1} працює $O(f(n) + g(n))$. Отже $f(n) + g(n)$ конструктивною.

2. $h_2 = f(n)g(n)$

   Будуємо аналогічну пунту 1 МТ, окрім:

   На кінець машина записує на вихід слово $1^{f(n)}$, отримане M_f, $g(n)$ разів, пробігаючи слово $1^{f(n)}$ $g(n)$ разів. Отримуємо час роботи $O(f(n)g(n))$

3. $h_3 = g(f(n))$

   Будуємо машину Тюрінга аналогічну пункту 1, окрім того, що підмашині M_g на вхід буде пдаватися слово не з першої стрічки, а зі стрічки l_f. Тобто ця підмашина буде працювати на l_f - $l_f + l_g - 2$ стрічках.

4. $h_4 = 2^{f(n)}$

   Є композицією конструктивних функцій, що є конструктивною функцією.

5. $h_5 = f(n)^{g(n)}$

   Будуємо машину Тюрінга аналогічно, але з правилами як у задачі 2.6.4. На вихідну стрічку записується слово з додаткової $f(n) - 1$ разів.

   Таким чином $f(n)^n$ буде конструктивною за часом, а отже конструктивною буде і $f(n)^{g(n)}$.

9 Problem 2.11

Візьмемо на приклад машину Тюрінга з задачі 2.6.2. Відкинемо третю стрічку, додаткова буде вихідною. Залишається лише перетворення числа з унарного на бінарний запис. Замінюємо всі нулі на одиниці та отримуємо 1^{logn}. Пам'яті використано $\lceil logn \rceil$