1 Поняття про алгоритмічні системи.

Нехай $A = \langle \varphi, \Pi \rangle$ - деякий алгоритм із системою правил Π.

Алгоритм $A_F = \langle \varphi, \Pi_F \rangle$, де Π_F формальний опис системи Π, називатимемо формальним еквівалентом алгоритму A.

Алгоритмічною системою називають спосіб задання алгоритмів у деякому фіксованому формалізмі F, який дає змогу для довільного наперед заданого алгоритму A задати його формальний еквівалент A_F.

Алгоритмічні системи можна вважати формалізацією поняття алгоритму. Виділяють три типи таких систем, які відрізняються початковими евристичними міркуваннями стосовно того, що таке алгоритм:

- у першому типі поняття алгоритму пов'язане з найбільш традиційними поняттями математики - обчисленнями та числовими функціями (рекурсивні функції).
- другий тип ґрунтується на уявленні про алгоритм як детермінований пристрій, здатний виконувати в кожний окремий момент лише дуже примітивні операції (машина Тьюрінга)
- третій тип алгоритмічних систем - перетворення слів у довільних алфавітах (канонічна система Поста й нормальні алгоритми Маркова).

Один з фундаментальних результатів теорії алгоритмів є такий: усі алгоритмічні системи є рівносильними в тому сенсі, що в кожній з них описують один і той самий клас алгоритмів.

Кожна алгоритмічна система охоплює об'єкти двох категорій - елементарні оператори та елементарні розпізнавачі.

Елементарні оператори - це достатньо прості алфавітні оператори. За допомогою послідовного виконання таких операторів відбувається реалізація довільних алгоритмів у заданій алгоритмічній системі.

Елементарні розпізнавачі використовують для розпізнавання певних властивостей інформації, перетворюваної алгоритмом і для зміни послідовності виконання елементарних операторів.

Для задання допустимих у системі елементарних операторів і розпізнавачів кожна система використовує свої засоби.

② Метод „Поділяй і володарюй"

Алгоритми вигляду „поділяй і володарюй" (ПВ) мають рекурсивну структуру: для розв'язування задачі вони рекурсивно викликають самі себе один чи декілька разів, щоб розв'язати допоміжну задачу, яка безпосередньо стосується сформульованої.

Такі алгоритми часто розробляють за допомогою методу часткових цілей: складну задачу розбивають на декілька простіших, які подібні до вихідної задачі, але мають менший обсяг; далі ці допоміжні задачі розв'язують рекурсивним методом, після чого отримані розв'язки комбінують, щоб отримати розв'язок вихідної задачі.

У методі „поділяй і володарюй" на кожному рівні рекурсії виділяють три етапи.

1. Поділ задач на декілька підзадач.
2. Рекурсивне розв'язування цих підзадач. Коли обсяг підзадачі достатньо малий, виділені підзадачі розв'язують безпосередньо.
3. Комбінування розв'язку вихідної задачі з розв'язків допоміжних задач.

```
Поділяй_І_Володарюй (дані, N, розв'язок)
  якщо (N <= граничний розмір)
      тоді Прямий_Розв'язок (дані, N, р-ок)
  в іншому випадку
      Поділ_Даних (дані, N, підмножини, розмПідмн., M)
  для i=1 до M
      повторювати Поділяй_Володарюй (підм[i],
          розмПідмн[i], підрозв[i])
      Комбінація_Розв'язків (підрозв. M, розв'язок)
```

$$T_{ПВ}(N) = \begin{cases} T_{ПР}(N) \longrightarrow \text{при } N < \text{граничнийРозмір} \\ T_{ПД}(N) + \sum_{i=1}^{M} T_{ПВ}(\text{розмПідмножини}[i]) + T_{КР}(N), \text{ в ін.в.} \end{cases}$$

③ Створити машину Тьюрінга для обчислення ф-ції $L_4^5(x_1, x_2, x_3, x_4, x_5) = x_4$, якщо x_1, x_2, x_3, x_4, x_5 – числа записані в алфавіті $A = \{|\}$, а між числами x_1, x_2, x_3, x_4, x_5 ставиться $*$.

	$\vert$	$*$	Λ
q_0	ΛR	$\Lambda q_1 R$	R
q_1	ΛR	$\Lambda q_2 R$	
q_2	ΛR	$\Lambda q_3 R$	
q_3	R	$\Lambda q_4 R$	
q_4	ΛR		Λq_F

Ідея: п'ять чисел записані по порядку через $*$. Витерти перші три із $*$ відповідно, а 4 залишити. Після 4 витерти 5 число і $*$. Тобто залишити лише 4 число.

$| * | * | * | * | \;\to\; \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda | \Lambda\Lambda$

$\underset{q_0}{|} * | * | * | * | \to \Lambda \underset{q_0}{*} | * | * | * | \to \Lambda\Lambda \underset{q_1}{|} * | * | * | \to$

$\to \Lambda\Lambda\Lambda \underset{q_1}{*} | * | * | \to \Lambda\Lambda\Lambda\Lambda \underset{q_2}{|} * | * | \to \Lambda\Lambda\Lambda\Lambda\Lambda \underset{q_2}{*} | * | \to$

$\to \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda \underset{q_3}{|} * | \to \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda | \underset{q_3}{*} | \to \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda | \Lambda \underset{q_4}{|} \to$

$\to \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda | \Lambda\Lambda \underset{q_F}{\Lambda}$