

Іспит із теорії алгоритмів
студента групи ПМІ-21
Дзисюка Ярослава

Білет № 10

Завдання 1

Оператор примитивної рекурсії.

Розглянемо функції $g(x_1, \dots, x_n)$ з кількістю параметрів n та функцію $h(x_1, \dots, x_n, y, z)$ з кількістю параметрів $n+2$.

Нехай виконується

$$\begin{cases} f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n); \\ f(x_1, \dots, x_n, y+1) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)) \end{cases}$$

Потді кажуть, що функція f отримана із функцій g і h із застосуванням оператора примитивної рекурсії.

Подібно, при нульовому аргументі ми використовуємо перший рядок, а при будь-якому ненульовому – використовуємо другий, так як у нас множина натуральних чисел виходить з нуля.

Завдання 2

Методи розробки ефективних алгоритмів.

Динамічне програмування

а) До фундаментальних методів розробки алгоритмів можна віднести:

— Метод часткових цілей. Суть методу полягає у тому, щоб замість розв'язування складної задачі звести її розв'язок до вирішення послідовності простіших задач.

У метода є і мінус, а саме, що не існує універсальних правил для визначення типу задачі, щоб цю задачу можна було невипро розв'язати за допомогою методу часткових цілей, крім того цей метод не завжди легко перенести на конкретну задачу.

— Метод підйому. Суть методу: приймаємо або обираємо поточковий розв'язок задачі. А далі ми починаємо рухатися до кращих та оптимальніших розв'язків

даній задачі. Цей процес поліпшення називають підходом, від того і назва методу. Зупинимося підійматися, коли оптимальні розв'язки закінчуються. Плюсом методу є його користь, коли потрібно швидко отримати наближений розв'язок, мінусом — немає гарантій, що кінцевий розв'язок буде оптимальним.

— **Рекурсія**. Це процедура, яка звертається сама до себе. Вона призводить у поєднанні з іншими методами до ефективніших розв'язків.

— **Метод відпрацювання назад**. Використовується для розв'язування різноманітних головоломок. Суть: починають із розв'язку задачі і рухаються до формулювання, тобто ззаду наперед. У випадку, якщо дії оборотні, рухаються знову у зворотньому напрямку — від формулювання до розв'язку.

— **Метод "Поділяй і володарюй"**. У ньому наявна рекурсивна структура, щоб розв'язати

допоміжну задачу, яка стосується пошуків.
Метод можна поділити на етапи: 1) поділ задачі на підзадачі (тут часто використовується метод гасових цілей), 2) рекурсивне виконання підзадач, 3) комбінування розв'язку вихідної та допоміжних задач.

б) Динамічне програмування. Це поняття використовується для характеристики алгоритмів, які діють в умовах відомості від об'єкту, від чого і назва - динамічне. Динамічне програмування використовується до задач, де щоб отримати оптимальний розв'язок, слід зробити певний набір виборів. Ефективніше, коли допоміжна підзадача виникає унаслідок декількох варіантів виборів. Ми здіємо розв'язок підзадачі, щоб не розв'язувати її знову спочатку, ми її самі покращуємо час роботи алгоритму.

Динамічне програмування можна поділити на етапи:

1. Опис оптимального розв'язку
2. Використання рекурсії для пошуку значення, що підпадає під оптимальне вирішення.
3. Обчислення значення, що підпадає під оптимальний розв'язок.

Додатково ми можемо також отримати процес утворення оптимального вирішення задачі.

Завдання 3

Задавши алфавіт $A = \{k, e, m, n\}$.

Скласти нормальний алгоритм, який всі літери m і n замінює порожнім символом, перше входження ll замінити на nnn .

Розв'язок:

$m \rightarrow \Lambda$ $ll \rightarrow nnn$.

$n \rightarrow \Lambda$, де Λ - порож. символ.

$P_1: m \rightarrow \Lambda$

$P_2: n \rightarrow \Lambda$

$P_3: ll \rightarrow nnn$