

## Лекція №6

# **ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ**

## Граф алгоритму

Будь-яка програма для "звичайного комп'ютера" описує деяку родину алгоритмів. Вибір конкретного алгоритму при її реалізації визначається тим, як "спрацьовують" умовні оператори, які залежать від вхідних даних. Тому "звичайний" комп'ютер завжди виконує деяку послідовність дій, яка *однозначно* визначається програмою та вхідними даними, причому в кожний момент часу виконується рівно одна дія.

Інакша ситуація в системах з паралельною архітектурою. Для них в кожний момент часу може виконуватися цілий набір операцій, які не залежать одна від іншої. На довільній конкретній паралельній системі ці набори та послідовність їх виконання *однозначно* визначаються програмою та вхідними даними. На різних системах ці набори та послідовності можуть бути *різними*. Тим не менш, для гарантування отримання однакового результату порядок виконання послідовності операцій має задовольняти певні умови.

Деякі операції алгоритму використовують результати виконання інших операцій і тому *обов'язково мають виконуватися* після цих операцій. Таким чином на множині операцій розробник програми явно чи неявно задає деякий "частковий порядок", який для довільних двох операцій вказує, яка з них має виконуватися раніше, або констатує, що операції можуть виконуватися незалежно.

Кожній операції алгоритму ставиться у відповідність *вершина графа*. Вершина графа з'єднується дугою з іншою вершиною, якщо перша вершина безпосередньо передує іншій (тобто результат першої операції є аргументом другої операції). Отриманий граф

У множині вершин графа також виділяють дві групи вершин: *вхідні вершини*, у які не входить жодна дуга, та *вихідні вершини*, з яких не виходять дуги.

Граф алгоритму майже завжди залежить від вхідних даних. Якщо навіть у програмі відсутні умовні оператори, то він залежить від розмірів вхідних (вихідних) масивів, бо вони визначають загальну кількість операцій, а отже, і вершин графа. Таким чином на практиці майже завжди мають справу з алгоритмами, граф яких є параметризованим. Від значення параметрів залежить не тільки кількість вершин, а й уся сукупність дуг.

Якщо програма не містить умовних операторів, то її саму і також алгоритм, який вона реалізовує, називається *детермінованим*.

Існує принципова відмінність між детермінованими та недетермінованими алгоритмами. Для детермінованого алгоритму завжди існує взаємно однозначна відповідність між всіма операціями та усіма вершинами графу алгоритму незалежно від значення вхідних параметрів. Для недетермінованого алгоритму такої відповідності нема.

Надалі, будуть розглядатися лише детерміновані алгоритми. Їх аналіз є простішим, ніж у загальному випадку. Крім того, багато недетермінованих алгоритмів є "майже" детермінованими, тобто зводяться до детермінованих.

Уведений у розгляд граф алгоритму є орієнтованим ациклічним мультиграфом. Його ациклічність впливає із того, що у довільних програмах реалізують лише явні обчислення і ніяка величина не визначається сама через себе.

## Твердження 6

Нехай  $G=(V,E)$  — зв'язний ациклічний граф, який має  $n$  вершин. Тоді існує таке  $k \leq n$  число  $k$  що усі вершини графа можна так помітити одним із індексів  $1, 2, \dots, k$ , що якщо дуга виходить із вершини з індексом  $i$  та входить у вершини з індексом  $j$ , то  $i < j$ .

Граф, отриманим у відповідності із твердженням 1, називається *строгою паралельною формою* графа алгоритму. Група вершин, які мають однакові індекси називається *ярусом* паралельної форми, а кількість вершин у групі — *шириною* ярусу. Кількість ярусів у паралельній формі називається *висотою* паралельної форми, максимальна ширина ярусу — її *шириною*. Відповідні "ботанічні" терміни застосовуються також і безпосередньо до алгоритмів чи програм.

Серед строгих паралельних форм існує форма, у якій вхідні вершини мають індекс 1, а довжина усіх шляхів від початкових вершин до вершин індексу  $k$  рівна  $k - 1$ . Така форма графа називається *канонічною паралельною формою*. Для заданого графа його канонічна форма єдина.



Якщо для простоти вважати, що усі операції алгоритму виконуються за одиницю часу, то висота паралельної форми алгоритму рівна часу реалізації алгоритму. Якщо алгоритм реалізовується на "звичайному" комп'ютері, то усі яруси паралельної форми містять одну вершину. Така паралельна форма називається лінійною.

Показано [1], що незалежно від того, яка паралельна форма алгоритму реалізується на комп'ютері, результат реалізації буде одним і тим самим.

# Твердження 7

Нехай при виконанні операцій помилки заокруглень визначаються лише значеннями аргументів. Тоді при одних і тих самих вхідних даних усі реалізації алгоритму, які відповідають одному і тому самому частковому порядку на операціях, дають однаковий результат включно із помилками заокруглень.

# Задача

Зобразити граф алгоритму обчислення виразу

$$S = a_1 + a_1 \cdot a_2 + \dots + a_1 \cdot a_2 \cdot \dots \cdot a_6,$$

вказати його ширину, висоту та знайти завантаженість систему у випадку реалізації алгоритму на системі, яка складається із:

1. Одного процесора.
2. Трьох процесорів.
3. 16 процесорів.

Дякую за увагу!