

Лекція 2. Поняття алгоритму

Алгоритм (латинізов. *Algorithmi*, від імені перського математика IX ст. **аль-Хорезмі**) – послідовність, система, набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу завдань після скінченного числа операцій. При написанні комп'ютерних програм алгоритм описує логічну послідовність операцій. Для візуального зображення алгоритмів часто використовують блок-схеми.

Єдиного «вірного» визначення поняття «алгоритм» немає.

«Алгоритм – це кінцевий набір правил, що визначає послідовність операцій для рішення конкретної множини задач і володіє п'ятьома важливими рисами: кінцевість, визначеність, введення, виведення, ефективність». (Д. Э. Кнут)

«Алгоритм – це всяка система обчислень, виконуваних по строго певних правилах, що після якого-небудь числа кроків свідомо приводить до рішення поставленого завдання». (А. Колмогоров)

«Алгоритм – це точне приписання, що визначає обчислювальний процес, що йде від варіюємих вихідних даних до шуканого результату». (А. Марков)

«Алгоритм – точне приписання про виконання в певному порядку деякої системи операцій, що ведуть до рішення всіх завдань даного типу». (Філософський словник / Під ред. М. М. Розенталя)

«Алгоритм – строго детермінована послідовність дій, що описує процес перетворення об'єкта з початкового стану в кінцеве, записана за допомогою зрозумілих виконавцеві команд». (Микола Дмитрович Угринович, підручник «Інформатика й информ. технології»)

Походження терміну алгоритм

Слово *алгоритм* походить від імені перського вченого, астронома та математика Аль-Хорезмі. Приблизно 825 до н. е. він написав трактат, в якому описав придуману в Індії позиційну десяткову систему числення.

В першій половині XII століття книжка потрапила до Європи в перекладі латинською мовою під назвою *Algorithmi de numero Indorum*. Вважається, що перше слово в перекладі відповідає невдалій латинізації імені Аль-Хорезмі, а назва перекладу звучить як «Алгоритмі про індійську лічбу».

Через невірне тлумачення слова *Algorithmi* як іменника в множині ним стали називати *метод обчислення*.

Неформальне визначення

Кожен алгоритм передбачає існування *початкових* (вхідних) даних та в результаті роботи призводить до отримання певного *результату*. Робота кожного алгоритму відбувається шляхом виконання послідовності деяких елементарних дій. Ці дії називають *кроками*, а процес їхнього виконання називають *алгоритмічним процесом*. В такий спосіб відзначають властивість *дискретності алгоритму*.

Важливою властивістю алгоритмів є *масовість*, або можливість застосування до різних вхідних даних. Тобто, кожен алгоритм покликаний розв'язувати клас однотипних задач.

Необхідною умовою, яка задовольняє алгоритм, є *детермінованість*, або *визначеність*. Це означає, що виконання команд алгоритму відбувається у єдиний спосіб та призводить до однакового результату для однакових вхідних даних.

Вхідні дані алгоритму можуть бути обмежені набором *припустимих вхідних даних*. Застосування алгоритму до неприпустимих вхідних даних може призводити до того, що алгоритм ніколи не зупиниться, або потрапить в тупиковий стан (зависання) з якого не зможе продовжити виконання.

Формальне визначення

Різноманітні теоретичні проблеми математики та прискорення розвитку фізики та техніки поставили на порядок денний точніше визначення поняття алгоритму.

Перші спроби уточнення поняття алгоритму та його дослідження здійснювали в першій половині XX століття Алан Тюринг, Еміль Пост, Жак Ербран, Курт Гедель, Андрій Марков, Алонзо Черч. Було розроблено декілька означень поняття алгоритму, але згодом було з'ясовано, що всі вони визначають одне й те саме поняття (див. теза Черча).

Машина Тюринга

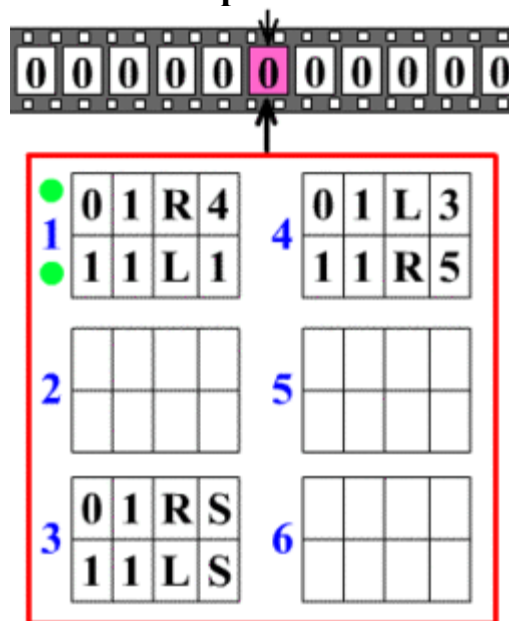


Рисунок 1 - Схематична ілюстрація роботи машини Тюринга

Основна ідея, що лежить в основі машини Тюринга, дуже проста.

Машина Тюринга - це абстрактна машина (автомат), що працює зі стрічкою окремих комірок, в яких записано символи.

Машина також має голівку для запису та читання символів із комірок, яка може рухатись вздовж стрічки. На кожному кроці машина зчитує символ з комірки, на яку вказує голівка, та, на основі зчитаного символу й внутрішнього стану робить наступний крок. При цьому, машина може змінити свій стан, записати інший символ в комірку, або пересунути голівку на одну комірку ліворуч або праворуч.

На основі дослідження цих машин було висунуто *тезу Тюринга* (основна

гіпотеза алгоритмів):

Для знаходження значень функції, заданої в деякому алфавіті, тоді і лише тоді існує деякий алгоритм, коли функція обчислювана за Тюрингом, тобто, коли її можна обчислити на придатній машині Тюринга.

Ця теза є аксіомою, постулатом, і не може бути доведена математичними методами, оскільки алгоритм не є точним математичним поняттям.

Алгоритмічно нерозв'язні задачі

Функцію f називають *обчислюваною* (англ. *computable*), якщо існує машина Тюринга, яка обчислює значення f для всіх елементів множини визначення функції. Якщо такої машини не існує, функцію f називають *необчислюваною*. Функція вважатиметься необчислюваною навіть, якщо існують машини Тюринга, здатні обчислити значення для підмножини з усієї множини вхідних даних.

Випадок, коли результатом обчислення функції f є булеве значення *істина* або *неправда* (або множина $\{0, 1\}$) називають *задачею*, яка може бути *розв'язною*, або *нерозв'язною* в залежності від обчислюваності функції f .

Важливо точно вказувати припустиму множину вхідних даних, оскільки задача може бути розв'язною для однієї множини та нерозв'язною для іншої.

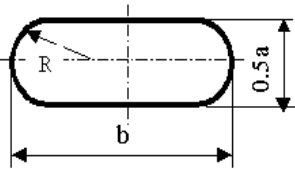
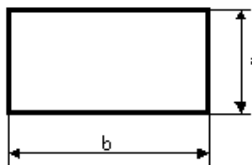
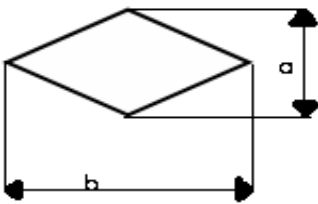
Однією з перших задач, для якої було доведено нерозв'язність є *проблема зупинки*. Формулюється вона наступним чином:

Маючи опис програми для машини Тюринга, визначити, чи завершить роботу програма за скінченний час, чи працюватиме нескінченно, отримавши будь-які вхідні дані.

Способи запису алгоритмів:

1. графічний (блок-схеми)
2. мовний
 - звичайна мова (напр., кулінарний рецепт)
 - псевдокод

Основні елементи схем алгоритму

Найменування	Позначення	Функція
Термінатор		Елемент відображає вхід у програму або вихід з неї (найбільш часте застосування - початок і кінець програми). Всередині фігури записується відповідна дія.
Процес		Виконання однієї або декількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.
Рішення		Показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення розумів, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить

		зазвичай у верхню вершину елемента. Якщо виходів два чи три те зазвичай кожен вихід позначається лінією, що виходить з решти вершини (бічних і нижній). Якщо виходів більше трьох, те їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.
Зумовлений процес		Символ відображає виконання процесу, що складається з однієї або декількох операцій, що визначений в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.
Дані		Перетворення даних у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).
Підготовка		Символ відображає модифікацію команди або групи команд з метою впливу на деяку наступну функцію (встановлення перемикача, модифікація індексного регістру або ініціалізація програми)
З'єднувач		Символ відображає вихід у частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.
Коментар		Використовується для більш детальної інформації про кроки, процесу або групи процесів. Опис поміщається з боку квадратної дужки і охоплюється їй по всій висоті. Пунктирна лінія йде до описуваного елемента, або групі елементів (при цьому група виділяється замкнутої пунктирною лінією). Також символ коментаря слід використовувати в тих випадках, коли обсяг тексту в будь-якому іншому символі (наприклад, символ процесу, символ даних та ін) перевищує його обсяг.

Псевдокод – компактна (найчастіше неформальна) мова опису алгоритмів, що використовує ключові слова імперативних мов програмування, але опускає

несуттєві подробиці й специфічний синтаксис.

Псевдокод звичайно опускає деталі, несуттєві для розуміння алгоритму людиною. Такими несуттєвими деталями можуть бути опис змінних, системно-залежний код і підпрограми. Головна мета використання псевдокоду – забезпечити розуміння алгоритму людиною, зробити опис більш сприйманим, ніж сирцевий код мовою програмування. Псевдокод широко використовується в підручниках і науково-технічних публікаціях, а також на початкових стадіях розробки комп'ютерних програм.

Базові керуючі структури

Назва структури	Псевдокод
присвоювання, введення, виведення	змінна = 0, введення (змінна), виведення (змінна)
розгалуження	якщо умова то (серія1 інакше серія 2)
цикл ПОКИ	поки умова поч серія кін

Приклад програми «Hello, world!»

алг HELLOWORLD

початок

виведення ('Hello,World!')

кінець алг HELLOWORLD

Загальний вид алгоритму:

алг назва алгоритму (аргументи й результати)

дано умови застосовності алгоритму

треба ціль виконання алгоритму

нач опис проміжних величин

| послідовність команд (тіло алгоритму)

кін

алг Сума квадратів (**арг** цілий n , **рез** цілий S)

дано | $n > 0$

треба | $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

поч цілий i

введення n ; $S:=0$

пч для i від 1 до n

$S:=S+i*i$

кц

вивід " $S =$ ", S

кін