МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"



Звіт

з лабораторної роботи № 1

З дисципліни «Програмування , частина 2»

Виконав:

Ст. гр. ТР-14

Бідний Д. С.

Прийняла:

Асистент Каф. ТК

Гордійчук-Бублівська О. В.

Львів - 2023

**Мета роботи:** ознайомитися із загальними принципами розробки алгоритмів, навчитися виконувати задачі щодо побудови різних типів алгоритмів.

**Теоретичні відомості**

Поняття алгоритму інтуїтивно зрозуміло та часто використовується в математиці та комп'ютерних науках. Говорячи неформально, алгоритм - це довільна коректно визначена обчислювальна процедура, на вхід якої подається деяка величина або набір величин, а результатом виконання якої є вихідна величина або набір значень.

Таким чином, алгоритм є послідовністю обчислювальних кроків, які перетворюють вхідні величини у вихідні. Алгоритм можна також розглядати як інструмент, який призначений для вирішення коректно поставленої обчислювальної задачі. У постановці задачі в загальних рисах визначаються відношення між входом та виходом.

В алгоритмі описується конкретна обчислювальна процедура, за допомогою якої можна досягнути виконання вказаних відношень. Можна навести загальні риси алгоритму:

• а. Дискретність інформації. Кожний алгоритм працює із даними: вхідними, проміжними, вихідними. Ці дані представляються у вигляді скінченних слів деякого алфавіту.

• b. Дискретність роботи алгоритму. Алгоритм виконується по кроках та при цьому на кожному кроці виконується тільки одна операція.

• c. Детермінованість алгоритму. Система величин, які отримуються в кожний (не початковий) момент часу, однозначно визначається системою величини, які були отримані в попередні моменти часу.

• d. Елементарність кроків алгоритму. Закон отримання наступної системи величин з попередньої повинен бути простим та локальним.

• e. Виконуваність операцій. В алгоритмі не має бути не виконуваних операцій. Наприклад, неможна в програмі призначити значення змінній 2 «нескінченність», така операція була би не виконуваною. Кожна операція опрацьовує певну ділянку у слові, яке обробляється.

• f. Скінченність алгоритму. Опис алгоритму повинен бути скінченним.

• g. Спрямованість алгоритму. Якщо спосіб отримання наступної величини з деякої заданої величини не дає результату, то має бути вказано, що треба вважати результатом алгоритму.

• h. Масовість алгоритму. Початкова система величин може обиратись з деякої потенційно нескінченної множини. Розглянемо для прикладу задачу сортування послідовності чисел у зростаючому порядку. Ця задача часто виникає на практиці і, фактично, буде центральною проблемою першого розділу даного курсу.

Задача сортування визначається формально наступним чином. Вхід: послідовність n чисел. Вихід: перестановка вхідної послідовності таким чином, що для всіх її членів виконується співвідношення. Наприклад, якщо на вхід подається послідовність , то вивід алгоритму сортування повинен бути таким: .

Подібна вихідна послідовність називається екземпляром задачі сортування. Взагалі, екземпляр задачі складається із входу, який необхідний для розв' язання задачі та який задовольняє усім обмеженням, які присутні в постановці задачі.

В комп'ютерних науках сортування є основною операцією (у багатьох програмах вона використовується в якості проміжного кроку), в результаті чого з'явилось багато якісних алгоритмів сортування.

Вибір найбільш адекватного алгоритму залежить від багатьох факторів, в тому числі й від кількості елементів для сортування, від їх порядку у вхідній послідовності, від можливих обмежень, які накладаються на членів послідовності. Кажуть, що алгоритм є коректним, якщо для кожного входу результатом його роботи є коректний вивід. Тоді коректний алгоритм розв'язує дану обчислювальну задачу.

Якщо алгоритм некоректний, то для деяких входів він може взагалі не завершити свою роботу або видати відповідь, яка відрізняється від очікуваної.

Для чого вивчати алгоритми? По-перше, алгоритми є життєво необхідними складовими для рішення будь-яких задач з різноманітних напрямків комп'ютерних наук. Алгоритми відіграють ключову роль у сучасному розвитку технологій. Тут можна згадати такі розповсюджені задачі, як:

- розв'язання математичних рівнянь різної складності, знаходження добутку матриць, обернених матриць;

- знаходження оптимальних шляхів транспортування товарів та людей;

- знаходження оптимальних варіантів розподілення ресурсів між різними вузлами (виробниками, верстатами, працівниками, процесорами тощо);

- знаходження в геномі послідовностей, які співпадають;

- пошук інформації в глобальній мережі Інтернет;

- прийняття фінансових рішень в електронній комерції;

- обробка та аналіз аудіо та відео інформації.

Цей список можна продовжувати й продовжувати і, власне кажучи, майже неможливо знайти таку галузь комп'ютерних наук та інформатики, де б не використовувались ті або інші алгоритми.

По-друге, якісні та ефективні алгоритми можуть бути каталізаторами проривів у галузях, які є на перший погляд далекими від комп'ютерних наук (квантова механіка, економіка та фінанси, теорія еволюції).

І, по-третє, вивчення алгоритмів це також цікавий процес, який розвиває математичні здібності та логічне мислення.

***Ефективність алгоритмів***. Припустимо, швидкодія комп'ютера та об'єм його пам'яті можна збільшувати до нескінченності. Чи була би тоді необхідність у вивченні алгоритмів? Так, але тільки для того, щоб продемонструвати, що метод розв'язку має скінченний час роботи і що він дає правильну відповідь. Якщо б комп'ютери були необмежено швидкими, підійшов би довільний коректний метод рішення задачі. Звісно, тоді найчастіше обирався би метод, який найлегше реалізувати.

Сьогодні є дуже потужні комп'ютери, але їх швидкодія не є нескінченно великою, як і пам'ять.

Таким чином, час обчислення - це обмежений ресурс, як і об'єм необхідної пам'яті. Цими ресурсами слід користуватись розумно, чому й сприяє застосування алгоритмів, які ефективні в плані використання ресурсів часу та пам'яті.

Алгоритми, які розроблені для розв'язання однієї та тієї самої задачі, часто можуть дуже сильно відрізнятись за ефективністю. Ці відмінності можуть бути набагато більше помітними, чим ті, які викликані застосуванням різного апаратного та програмного забезпечення.

Перший алгоритм, який буде розглядатись - сортування включенням, для своєї роботи вимагає часу, кількість якого оцінюється як 2 c1n , де n - розмір вхідних даних (кількість елементів у послідовності для сортування), с - деяка стала.

Цей вираз вказує на те, як залежить час роботи алгоритму від об'єму вхідних даних. У випадку сортування включенням ця залежність є квадратичною.

Використання коду, час роботи якого зростає повільніше, навіть при поганому комп'ютері та поганому компіляторі потребує на порядок менше процесорного часу. Для сортування 10 мільйонів чисел перевага сортування злиттям стає ще більш відчутною: якщо сортування включенням потребує для такої задачі приблизно 2,3 дня, то для сортування злиттям - менше 20 хвилин.

Загальне правило таке: чим більша кількість елементів для сортування, тим помітніша перевага сортування злиттям. Алгоритми, як і програмне забезпечення комп'ютера, являють собою технологію.

Загальна продуктивність системи настільки ж залежить від ефективності алгоритму, як і від потужності апаратних засобів.

Написанню програми на будь-якій мові програмування передує складання алгоритму, який показує послідовність виконання операцій, розгалуженість обчислювального процесу, логіку виконання всього ланцюгу програми від блоку введення вхідних даних до отримання кінцевого результату.

Існує чотири способи написання алгоритмів:

- вербальний (словесний);

-алгебраїчний (за допомогою літерно-цифрових позначень виконуваних дій);

- графічний;

- з допомогою алгоритмічних мов програмування.

Словесна форма запису алгоритмів використовується в різних інструкціях, призначених для виконання їх людиною.

Алгебраїчна форма найчастіше використовується у теоретичних дослідженнях фундаментальних властивостей алгоритмів.

Графічна форма відповідно до державних стандартів (ГОСТ) на оформлення документації прийнята як основна для опису алгоритмів.

Алгоритм записаний за допомогою алгоритмічної мови програмування називається програмою. Алгоритм у такій формі може бути введений у ЕОМ і після відповідного оброблення виконаний з метою отримання шуканого результату.

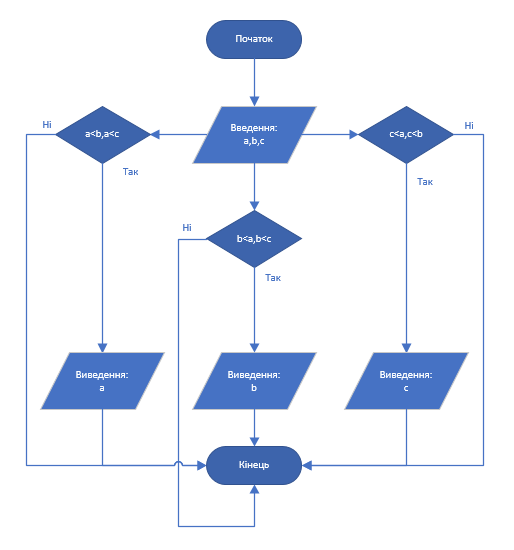
Блок-схеми – найбільш зручний спосіб візуального представлення алгоритмів. Для того, щоб не заплутатися в численних подробицях, є сенс складати блок-схему алгоритму в декілька ітерацій: почати з найбільш загального і поступово його уточнювати.

Блок-схема – наочне графічне зображення алгоритму, коли окремі його дії (етапи) зображуються за допомогою різних геометричних фігур (блоків), а зв’язки між етапами указуються за допомогою стрілок, що сполучають ці фігури.

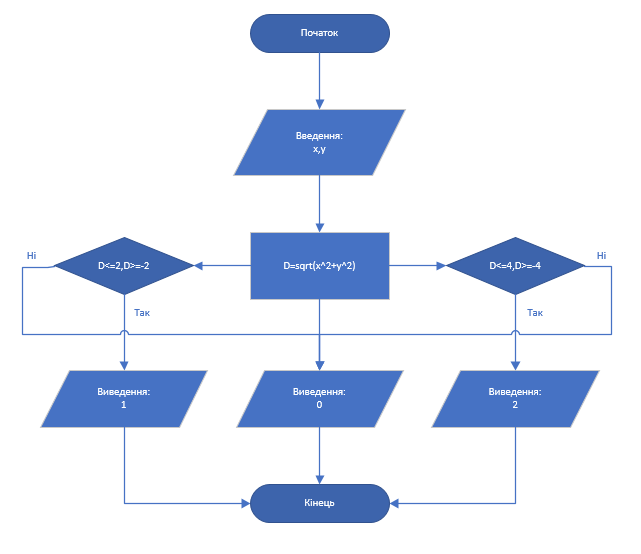
Блок-схеми відображають кроки, які повинні виконуватися комп’ютером, і послідовність їх виконання.

**Хід роботи**

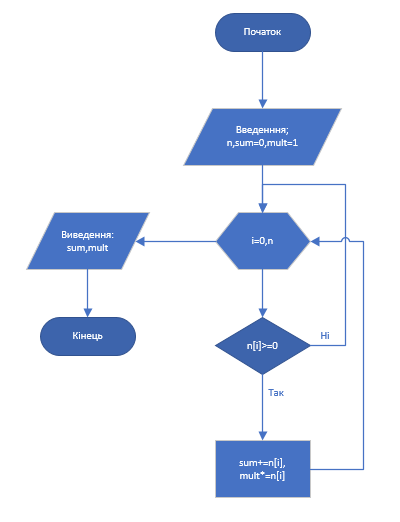
Завдання 2

****

Завдання 3



Завдання 4



**Висновок:** в ході лабораторної роботи я ознайомився із поняттям алгоритму, загальними принципами його написання і розробки, навчився виконувати задачі щодо побудови різних типів алгоритмів.