**Поняття алгоритму та його основні властивості**

**Алгоритм —** набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми.

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.

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.

**Основні властивості алгоритму.**

1. ***Визначеність (точність) :***кожна команда алгоритму має визначати однозначну дію виконавця.
2. ***Результативність*** ⎯ це означає, що алгоритм зводиться до виконання кінцевого числа дій і при цьому завжди є відповідь на запитання завдання .
3. ***Дискретність*** : алгоритм поділено на послідовність окремих кроків; тільки виконавши одну команду, виконавець зможе перейти до виконання наступної.
4. ***Зрозумілість*** : кожен алгоритм складається з орієнтацією на визначеного виконавця.
5. ***Масовість*** : алгоритм має забезпечувати розв’язання всього класу задач даного типу.

**Типи алгоритмів.**

* ***Лінійні*** ⎯ алгоритми, в яких дії виконуються послідовно без перевірки будь-яких умов.
* ***Розгалужені*** ⎯ алгоритми, в яких передбачені варіанти описів, в залежності від зміни умов (тобто перевірки умов “якщо - тоді ” в обов’язковому порядку).
* ***Циклічні*** ⎯ алгоритми, в яких окремі операції або групи операцій виконуються декілька разів. Їх ще називають алгоритмами з повтореннями.

**Хеш-таблиці. Головна ідея**

**Хеш-таблиця** — структура даних, що реалізує інтерфейс асоціативного масиву, а саме, вона дозволяє зберігати пари (ключ, значення) і здійснювати три операції: операцію додавання нової пари, операцію пошуку і операцію видалення за ключем.

Існує два основних варіанта Хеш-таблиць: з ланцюжками і з відкритою адресацією. Хеш-таблиця містить в собі деякий масив H, елементами якого є пари (Хеш-таблиця з відкритою адресацією) або списки пар (Хеш-таблиця з ланцюжками).

Виконання операцій в Хеш-таблиці починається з обчислення Хеш-функції від ключа. Отримане Хеш-значення i = hash(key) відіграє роль індексу в масиві H. Після цього операція (додавання, видалення, пошук) перенаправляється об'єктові, який зберігається у відповідній комірці масиву H[i].

Ситуація, коли для різних ключів отримується одне й те саме Хеш-значення, називається колізією. Такі події непоодинокі — наприклад, при додаванні в Хеш-таблицю розміром 365 комірок усього лише 23-х елементів ймовірність колізії вже перевищує 50 відсотків (якщо кожний елемент може з однаковою ймовірністю потрапити в будь-яку комірку). Через це механізм розв'язання колізій — важлива складова будь-якої Хеш-таблиці.

В деяких особливих випадках вдається взагалі уникнути колізій. Наприклад, якщо всі ключі елементів відомі заздалегідь (або дуже рідко змінюються), тоді для них можна знайти деяку досконалу Хеш-функцію, яка розподілить їх за комірками Хеш-таблиці без колізій. Хеш-таблиці, які використовують подібні Хеш-функції, не потребують механізму розв'язання колізій, і називаються Хеш-таблицями з прямою адресацією.

Важлива властивість Хеш-таблиць полягає в тому, що, при деяких розумних припущеннях, всі три операції (пошук, вставлення і видалення елементів) зазвичай виконується за час O(1). Але при цьому не гарантується, що час виконання окремої операції малий, з певною імовірністю час може бути сумірним із пошуком у списку. З ростом коефіцієнта заповнення таблиці ця імовірність, і, відповідно, середній час виконання операцій, ростуть. Тому при досягненні деякого значення коефіцієнта заповнення необхідно здійснювати перебудову індексу Хеш-таблиці: збільшити розміри масиву H і заново додати в порожню Хеш-таблицю всі пари.