**Швидке сортування**

**Швидке сортування** (англ. Quick Sort) — алгоритм сортування, добре відомий як алгоритм, розроблений Тоні Гоаром. Позаяк алгоритм використовує дуже прості цикли і операції, він працює швидше за інші алгоритми, що мають таку ж асимптотичну оцінку складності. Наприклад, зазвичай більш ніж удвічі швидший порівняно з сортуванням злиттям.

Метод відноситься до групи методів обміну, тобто до тієї ж, що і найповільніший - метод "бульбашки". Але корінна відмінність його полягає у організації обмінів між далекими за розташуванням елементами, що призводить до істотного прискорення роботи.

Ось викладення основної ідеї методу. Обирається деякий елемент, який назвемо медіаною. Усi елементи масиву на першому етапі розбирають на два підмасиви - елементів, більших від медіани та, відповідно, менших. На цьому, власне, перший етап закінчується. Далі процедура, що реалізує один етап, використовується рекурсивно, із застосуванням до обох частин масиву.

Робота продовжується до тих пір, поки у кожному з підмасивчиків, що на них розбивається масив, не стане лише по одному елементу. Це означатиме, що масив відсортовано.

Для організації роботи на першому етапі робиться інтуїтивне припущення, що якщо робити обміни не сусідніх елементів, як у методі "бульбашки", а самих дальніх одне від одного, то це має прискорити роботу програми.

Візьмемо за медіану елемент a[1]. Порівняймо його з останнім елементом a[N]! Якщо їх впорядковано, то a[1] порівнюємо з a[N-1], a[N-2], ..., поки не зустрінеться елемент, менший від медіани. Тоді їх слід поміняти місцями

І взагалі, після кожного обміну слід міняти "кінець" масиву, з яким порівнюється значення медіани. При цьому неаналізована частина завжди буде залишатися "всередині, "зліва" від неї будуть елементи, менші за медіану, а "справа" - більші.

Після того, як медіану порівняно з усіма елементами у масиві, вона, стрибаючи, стане на своє місце, тобто зліва від медіани стануть всі елементи, менші від неї, а справа - більші. Тоді надалі достатньо відсортувати лише праву та ліву частини масиву, бо положення медіани вже не мінятиметься.

**Динамічне програмування**

**Динамічне програмування** — розділ математики, який присвячено теорії і методам розв'язання багатокрокових задач оптимального управління.

У динамічному програмуванні для керованого процесу серед множини усіх допустимих управлінь шукають оптимальне у сенсі деякого критерію тобто таке яке призводить до екстремального (найбільшого або найменшого) значення цільової функції — деякої числової характеристики процесу. Під багатоступеневістю розуміють або багатоступеневу структуру процесу, або розподілення управління на ряд послідовних етапів (ступенів, кроків), що відповідають, як правило, різним моментам часу. Таким чином, в назві «Динамічне програмування» під «програмуванням» розуміють «ухвалення рішень», «планування», а слово «динамічне» вказує на суттєве значення часу та порядку виконання операцій в процесах і методах, що розглядаються.

Методи динамічного програмування є складовою частиною методів, які використовуються при дослідженні операцій, і використовуються як у задачах оптимального планування, так і при розв'язанні різних технічних проблем (наприклад, у задачах визначення оптимальних розмірів ступенів багатоступеневих ракет, у задачах оптимального проектування прокладення доріг та ін.)

Методи динамічного програмування використовуються не лише в дискретних, але і в неперервних керованих процесах, наприклад, в таких процесах, коли в кожен момент певного проміжку часу необхідно ухвалювати рішення. Динамічне програмування також дало новий підхід до задач варіаційного числення.

Хоча метод динамічного програмування суттєво спрощує вихідні задачі, та безпосереднє його використання, як правило, пов'язане з громіздкими обчисленнями. Для подолання цих труднощів розробляються наближені методи динамічного програмування.

Динамічне програмування є водночас і методом математичної оптимізації і методом комп'ютерного програмування. В обох контекстах воно використовує підхід спрощення пошуку розв'язку складної задачі, розбиттям її на простіші підзадачі, часто методом рекурсії. Хоча деякі задачі не можуть бути розв'язані таким чином, рішення, які охоплюють кілька точок у часі дійсно часто розбиваються рекурсивно на підзадачі. Белман називав це принципом оптимальності.