# **Метод декомпозиції. Сортування злиттям.**

# **Метод декомпозиції**

Метод декомпозиції також відомий як метод «розділяй і володарюй». В самій назві зберігається зміст алгоритмів, які базуються на цьому методі. Вони працюють за таким планом:

1. Екземпляр задачі розбивається на декілька менших екземплярів тієї ж задачі, в ідеалі – це екземпляри однакового розміру;
2. Вирішуються менші екземпляри задачі (зазвичай, рекурсивно, хоча для більших екземплярів іноді застосовується інший алгоритм);
3. За необхідності, вирішення задачі знаходиться шляхом комбінації вирішень менших екземплярів.

# **Сортування злиттям**

Алгоритм сортування, в основі якого лежить принцип «розділяй та володарюй». Його роботу можна описати такими частинами:

**Розподіл**: послідовність, що складається з n елементів , розбивається на дві менші послідовності, кожна з яких містить n / 2 елементів.

**Підкорення**: кожна з одержаних послідовностей сортується окремо, рекурсивно таким же алгоритмом.

**З'єднання** : злиття двох відсортованих послідовностей в один для отримання остаточного результату.

Рекурсія досягає своєї нижньої межі, коли довжина сортованої послідовності стає рівною 1. У цьому випадку вся робота вже зроблена, оскільки будь-яку таку послідовність можна вважати впорядкованою.

Основна операція, що проводиться в процесі сортування за алгоритмом злиття – це об'єднання двох відсортованих послідовностей в ході комбінування (останній етап). Це робиться за допомогою допоміжної процедури MERGE (A, p, q, r), де А - масив, а р, q і r - індекси, нумерують елементи масиву, так, що р <= *q < r*. У цій процедурі передбачається, що елементи підмасивів А [p, .. q] і A [q + 1, .r] впорядковані. Вона зливає ці два підмасива в один відсортований, елементи якого замінюють поточні елементи підмасива А [p, .. r].

Алгоритм дає хороші показники по швидкодії роботи, навіть у порівнянні з сортуванням методом бінарних дерев. Єдиний недолік – необхідність використовувати додаткової пам'яті –  додатковий масив того ж розміру, що сортується.

# **Псевдокод алгоритму**

Процедура MergeSort (A, p, q) M e r g e S o r t ( A , p , q ) {\displaystyle \;MergeSort(A,p,q)} здійснює часткове впорядкування масиву AA {\displaystyle \;A} , впорядковуючи його елементи з p-го по q-ий (MergeSort(A, 1, length(A)) ( M e r g e S o r t ( A , 1 , l e n g t h ( A ) ) {\displaystyle (\;MergeSort(A,1,length(A))} здійснить впорядкування всього масиву).

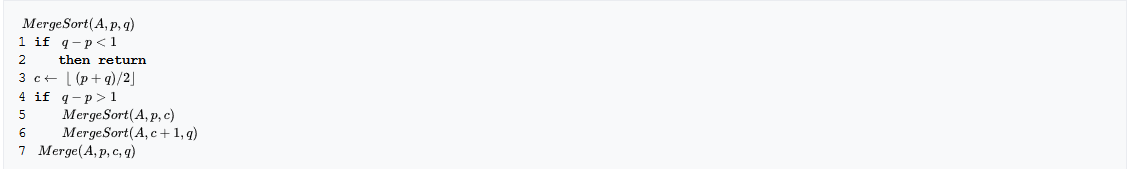


Рис. 1 – Процедура MergeSort

MergeSort використовує допоміжну процедуру Merge (A, p, c, q) M e r g e ( A , p , c , q ) {\displaystyle \;Merge(A,p,c,q)} , що здійснює об'єднання частин масиву A з p-го по c-ий елемент і з c+1-го по q-ий елемент в один впорядкований підмасив. Для цього використовується один додатковий масив B {\displaystyle \;B} B такої ж довжини як і A {\displaystyle \;A} A (у деяких реалізаціях B B {\displaystyle B\;} вдвічі коротший за A {\displaystyle A\;} A — мінімально можлива його довжина).

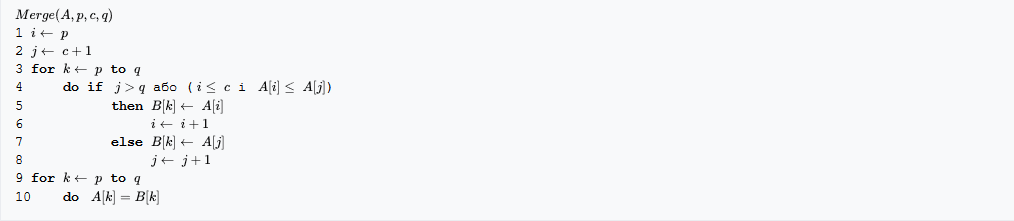


Рис. 2 – Процедура Merge

# **Складність алгоритму**

При кожному виконанні тіла циклу while значення всіх елементів масиву копіюються в допоміжний масив і назад по одному разу, тобто виконується O(n) елементарних дій. Після останнього k-го кроку 2k < 2n, тобто k < (1+logn), і це означає, що тіло циклу while виконується O(logn) разів. Отже, складність алгоритму оцінюється як **O(nlogn)**.

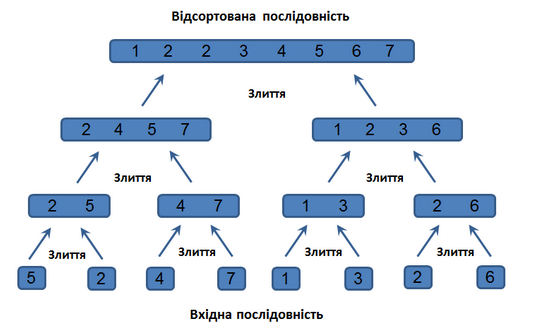


Рис. 3 – Сортування злиттям