**Просторово-часовий компроміс: сортування підрахунком**

Ідея сортування підрахунком полягає в тому, щоб підрахувати ля кожного елемента списку, що сортується, загальну кількість елементів, менших за даний, і занести отриманий результат у таблицю. Отримані числа вказують на розташування елемента у відсортованому списку. Таких алгоритм називається сортуванням підрахунком порівнянь.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Початковий масив | 62 | 31 | 84 | 96 | 19 | 47 |
| Початково | 0 | 0 | 0 | 0 | 0 | 0 |
| i=0 | 3 | 0 | 1 | 1 | 0 |  |
| i=1 |  | 1 | 2 | 2 | 0 | 1 |
| i=2 |  |  | 4 | 3 | 0 | 1 |
| i=3 |  |  |  | 5 | 0 | 1 |
| i=4 |  |  |  |  | 0 | 2 |
| Кінцевий стан | 3 | 1 | 4 | 5 | 0 | 2 |
| Кінцевий масив | 19 | 31 | 47 | 62 | 84 | 96 |

Часова ефективність даного алгоритму має бути квадратичною, оскільки алгоритм розглядає усі різні пари масиву з n елементів.

Ідея підрахунку добре спрацьовує в ситуації, коли сортовані елементи належать невеликій множині значень. Тобто, якщо значення елементів являють собою цілі числа в межах від *l* до, то *u* ми можемо вирахувати частоти появи кожного з цих значень та зберегти їх у масиві *F[0…u-l].* Далі перші *F[0]* позицій у відсортованому списку мають бути заповнені значенням *l*, наступні *F[1]* позиції – значенням *l+1*, і так далі.

У припущенні фіксованого діапазону значень, цей алгоритм лінійний, оскільки виконує два послідовних проходи по одному масиву *А.* Ефективність цього способу отримана завдяки конкретному виду вхідних даних, для яких, разом з використанням додаткової пам’яті, працює сортування підрахунком розподілу.