

Nama: Bariq Adyatma

Nim: 21091397019

Kelas: A

INSERTION SORT

Insertion Sort merupakan sebuah teknik pengurutan dengan cara membandingkan dan mengurutkan dua data pertama pada array, kemudian membandingkan data para array berikutnya apakah sudah berada di tempat semestinya. Algorithma insertion sort seperti proses pengurutan kartu yang berada di tangan kita. Algorithma ini dapat mengurutkan data dari besar ke kecil (Ascending) dan kecil ke besar (Descending). Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algorithma ini adalah $O(n^2)$ di mana n adalah jumlah item.

Berikut gambaran dari implementasi Insertion Sort:

1st Cycle:

(70, 60, 30, 50, 40, 20) -> (60, 70, 30, 50, 40, 20)
(60, 70, 30, 50, 40, 20)

2nd Cycle:

(60, 70, 30, 50, 40, 20) -> (60, 30, 70, 50, 40, 20)
(60, 30, 70, 50, 40, 20) -> (30, 60, 70, 50, 40, 20)
(30, 60, 70, 50, 40, 20)

3rd Cycle:

(30, 60, 70, 50, 40, 20) -> (30, 60, 50, 70, 40, 20)
(30, 60, 50, 70, 40, 20) -> (30, 50, 60, 70, 40, 20)
(30, 50, 60, 70, 40, 20) -> (30, 50, 60, 70, 40, 20)
(30, 50, 60, 70, 40, 20)

4th Cycle:

(30, 50, 60, 70, 40, 20) -> (30, 50, 60, 40, 70, 20)
(30, 50, 60, 40, 70, 20) -> (30, 50, 40, 60, 70, 20)
(30, 50, 40, 60, 70, 20) -> (30, 40, 50, 60, 70, 20)
(30, 40, 50, 60, 70, 20) -> (30, 40, 50, 60, 70, 20)
(30, 40, 50, 60, 70, 20)

5th Cycle:

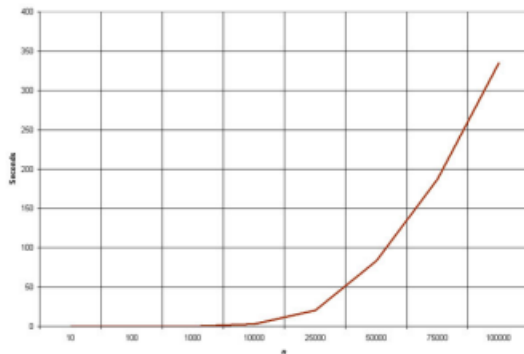
(30, 40, 50, 60, 70, 20) -> (30, 40, 50, 60, 20, 70)
(30, 40, 50, 60, 20, 70) -> (30, 40, 50, 20, 60, 70)
(30, 40, 50, 20, 60, 70) -> (30, 40, 20, 50, 60, 70)
(30, 40, 20, 50, 60, 70) -> (30, 20, 40, 50, 60, 70)
(30, 20, 40, 50, 60, 70) -> (20, 30, 40, 50, 60, 70)
(20, 30, 40, 50, 60, 70)

Kompleksitas Insertion Sort

Algoritma Insertion Sort juga terdiri dari 2 kalang bersarang. Dimana terjadi N-1 Pass (dengan N adalah banyak elemen struktur data), dengan masing-masing Pass terjadi i kali operasi perbandingan. i tersebut bernilai 1 untuk Pass pertama, bernilai 2 untuk Pass kedua, begitu seterusnya hingga Pass ke N-1.

$$T(n) = 1 + 2 + \dots + n - 1 = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = O(n^2)$$

Walaupun sama seperti dua algoritma sorting sebelumnya, insertion sort lebih mangkus. Perhatikan gambar berikut:



Gambar 5. Grafik Efisiensi *Insertion Sort*

Berdasarkan gambar, Insertion Sort lebih dari dua kali lebih mangkus daripada Bubble Sort dan 40% lebih cepat daripada Selection Sort. Insertion Sort lebih baik tidak digunakan untuk menangani struktur data dengan lebih dari 2000 elemen.

