

Algoritma dan Struktur Data

Shell Sort

Umi Sa'adah

Tita Karlita

Entin Martiana Kusumaningtyas

Arna Fariza

2021



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Sorting Algorithms

1. Selection
2. Insertion
3. Bubble
4. Shell
5. Merge
6. Quick

Definisi

- Metode ini disebut juga dengan metode pertambahan menurun (*diminishing increment sort*).
- Metode ini dikembangkan oleh Donald L. Shell pada tahun 1959, sehingga sering disebut dengan Metode Shell Sort.
- Metode ini mengurutkan data dengan cara **membandingkan** suatu data dengan data lain yang memiliki **jarak tertentu** – sehingga membentuk sebuah **sub-list**-, kemudian dilakukan penukaran bila diperlukan.

Definisi

- Jarak yang dipakai didasarkan pada *increment value* atau *sequence number k*.
- Misalnya Sequence number (k) yang dipakai adalah 5,3,1.
- Tidak ada pembuktian di sini bahwa bilangan-bilangan tersebut adalah sequence number terbaik.
- Setiap sub-list berisi setiap elemen ke-k dari kumpulan elemen yang asli.

Definisi

- Contoh: Jika $k = 5$ maka sub-list nya adalah sebagai berikut :
 - $s[0] \ s[5] \ s[10] \dots$
 - $s[1] \ s[6] \ s[11] \dots$
 - $s[2] \ s[7] \ s[12] \dots$
 - dst
- Begitu juga jika $k = 3$ maka sub-list nya adalah:
 - $s[0] \ s[3] \ s[6] \dots$
 - $s[1] \ s[4] \ s[7] \dots$
 - dst

Proses Shell Sort secara umum

1. Buatlah sub-list yang didasarkan pada jarak (Sequence number k) yang dipilih.
2. Urutkan masing-masing sub-list tersebut
3. Gabungkan seluruh sub-list

Let's see this algorithm in action

Proses Shell Sort

Urutkan sekumpulan elemen di bawah ini , misalnya diberikan sequence number $k = 5, 3, 1$

30	62	53	42	17	97	91	38
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

Proses Shell Sort k=5

30 62 53 42 17 97 91 38

Step 1: Buat sub list $k = 5$

S[0] S[5]

S[1] S[6]

S[2] S[7]

S[3]



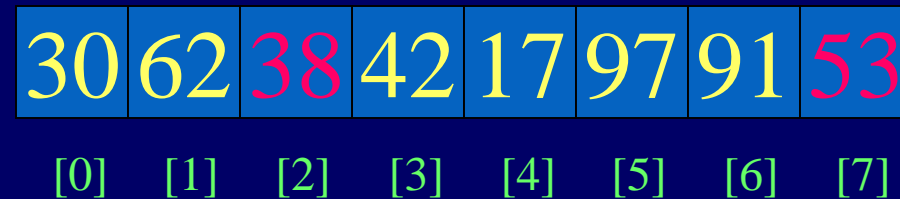
Step 2 - 3: Urutkan sub list & gabungkan

S[0] < S[5] This is OK

S[1] < S[6] This is OK

S[2] > S[7] This is not OK.

Swap them



Proses Shell Sort utk k=3

30 62 53 42 17 97 91 38

Step 1: Buat sub list $k = 3$

S[0] S[3] S[6]

S[1] S[4] S[7]

S[2] S[5]



Step 2 - 3: Urutkan sub list & gabungkan

S[0] S[3] S[6] 30, 42, 91 OK

S[1] S[4] S[7] 62, 17, 53 not OK

SORT them 17, 53, 62

S[2] S[5] 38, 97 OK



Shell Sort Process k=1

30 62 53 42 17 97 91 38

Step 1: Buat sub list **k = 1**

S[0] S[1] S[2] S[3] S[4] S[5] S[6] S[7]

30	17	38	42	53	97	91	62
----	----	----	----	----	----	----	----

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
							

Step 2 - 3: Urutkan sub list & gabungkan

Sorting akan seperti insertion sort

17	30	38	42	53	62	91	97
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

DONE

Pemilihan Sequence Number

- Disarankan jarak mula-mula dari data yang akan dibandingkan adalah: $N / 2$.
- Pada proses berikutnya, digunakan jarak $(N / 2) / 2$ atau $N / 4$.
- Pada proses berikutnya, digunakan jarak $(N / 4) / 2$ atau $N / 8$.
- Demikian seterusnya sampai jarak yang digunakan adalah 1.

Urutan prosesnya...

- Untuk jarak $N/2$:
 - Elemen pertama ($i=0$) dibandingkan dengan elemen dengan jarak $N/2$. Jika elemen pertama $>$ elemen ke $N/2$ maka kedua elemen tersebut ditukar.
 - Kemudian elemen kedua ($i=1$) dibandingkan dengan jarak yang sama yaitu $N/2$, yaitu elemen ke- $(N/2 + i)$.
 - Demikian seterusnya sampai seluruh elemen dibandingkan sehingga semua elemen ke- i selalu lebih kecil daripada elemen ke- $(N/2 + i)$.
- Ulangi langkah-langkah di atas untuk jarak $= N/4 \rightarrow$ lakukan perbandingan dan pengurutan sehingga semua elemen ke- i lebih kecil daripada elemen ke- $(N/4 + i)$.
- Ulangi langkah-langkah di atas untuk jarak $= N/8 \rightarrow$ lakukan perbandingan dan pengurutan sehingga semua elemen ke- i lebih kecil daripada elemen ke- $(N/8 + i)$.
- Demikian seterusnya sampai jarak yang digunakan adalah 1 atau elemen sudah terurut (`did_swap = false`).



Algoritma Metode Shell Sort

```
1.  Jarak ← N
2.  Selama Jarak > 1 do baris 3 s/d 15
3.      Jarak ← Jarak / 2.
4.      did_swap ← true
5.      Selama did_swap ← true kerjakan baris 6 s/d 14
6.          did_swap ← false
7.          i ← 0
8.          Selama i < (N - Jarak) kerjakan baris 9 s/d 13
9.              if Data[i] > Data[i + Jarak] do baris 10 dan 11
10.                  tukar(Data[i], Data[i + Jarak])
11.                  did_swap ← true
12.                  i ← i + 1
13.          End baris 8
14.      End baris 5
15.  End baris 2
```



Analisis Metode Shell Sort

- Running time dari metode Shell Sort bergantung pada pemilihan sequence number-nya.
- Disarankan untuk memilih sequence number dimulai dari $N/2$, kemudian membaginya lagi dengan 2, seterusnya hingga mencapai 1.
- Shell sort menggunakan 3 nested loop, untuk merepresentasikan sebuah pengembangan yang substansial terhadap metode insertion sort.

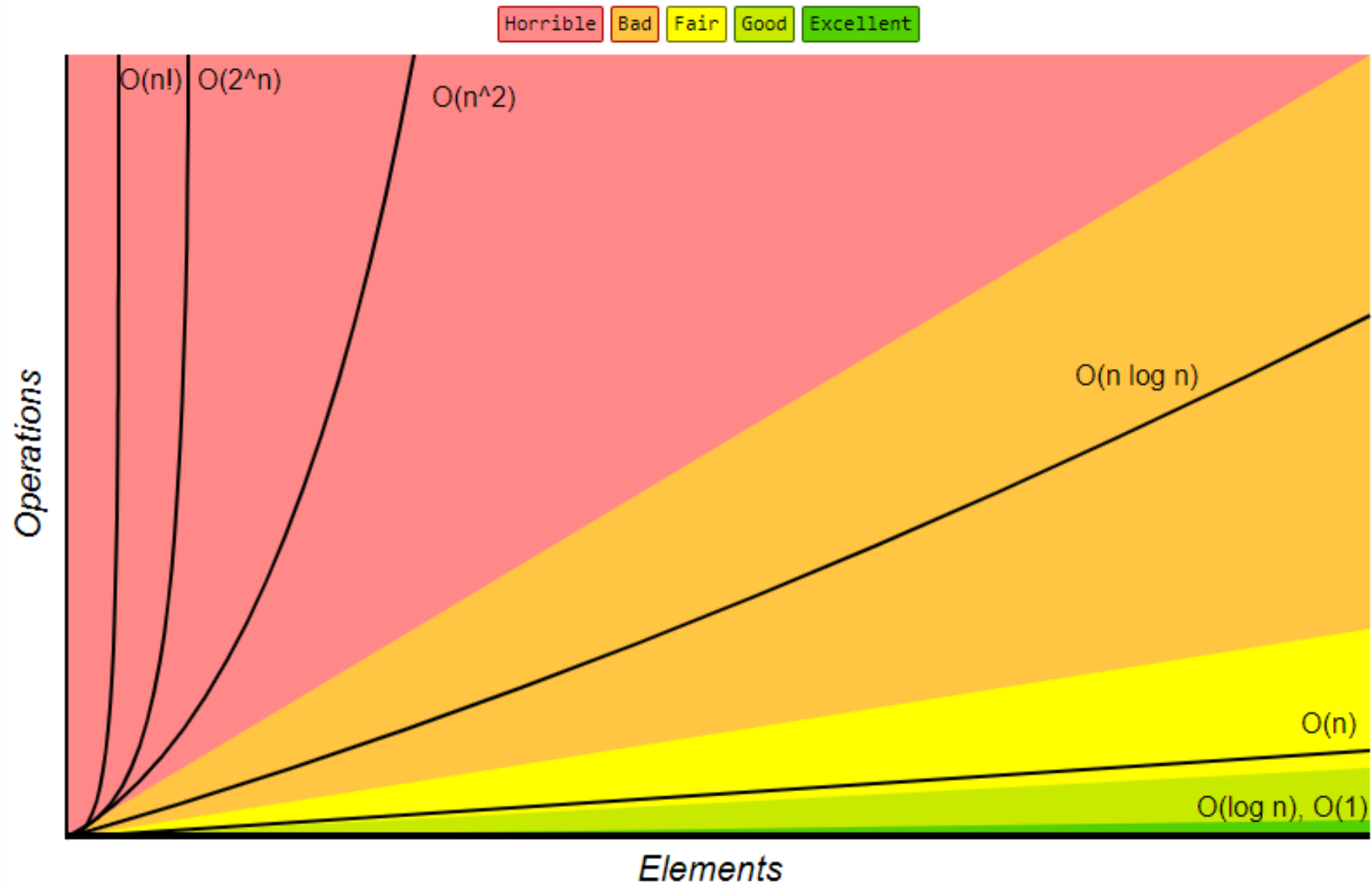
Perbandingan Running time (millisecond) antara Insertion and Shell

N	Insertion sort	Shell sort
1000	122	11
2000	483	26
4000	1936	61
8000	7950	153
16000	32560	358

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
<u>Quicksort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
<u>Mergesort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Timsort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Heapsort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
<u>Bubble Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Insertion Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Selection Sort</u>	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Tree Sort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(n)$
<u>Shell Sort</u>	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$
<u>Bucket Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
<u>Radix Sort</u>	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
<u>Counting Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
<u>Cubesort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$

Big-O Complexity Chart



Kesimpulan

- Algoritma Shell Sort mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu – sehingga membentuk sebuah sub-list-, kemudian dilakukan penukaran bila diperlukan.
- Merupakan pengembangan dari algoritma insertion short.

Latihan Soal

Urutkan data di bawah ini dengan Algoritma Bubble Sort dan Shell Sort, jelaskan pula langkah-langkahnya !

9 1 2 5 6 4