

Algoritme dan Struktur Data

Pencarian (*Searching*)

Putra Pandu Adikara

Universitas Brawijaya

Pencarian (Searching)

- Pada suatu data seringkali dibutuhkan pembacaan kembali informasi (*information retrieval*) dengan cara *searching*.
- Searching adalah proses pencarian data yang ada pada suatu deret data dengan cara menelusuri data-data tersebut.
- Tahapan paling penting pada *searching*: memeriksa jika data yang dicari sama dengan data yang ada pada deret data.

Algoritme Pencarian

- Macam algoritme pencarian:
 - Sequential Search
 - Binary Search
- Umumnya hasil dari metode pencarian memberikan nilai:
 - -1 bila tidak ditemukan
 - $x, \geq 0$ bila ditemukan, menghasilkan nilai $0 < x < n$
 x = indeks pada elemen *array*
 n = banyak elemen dalam *array*

1. Sequential Search

- Disebut juga *linear search* atau metode pencarian beruntun.
- Adalah suatu teknik pencarian data yang akan menelusuri tiap elemen satu per-satu dari awal sampai akhir.
- Data awal = tidak harus dalam kondisi terurut.

Algoritme Sequential Search

1. Input x (data yang dicari)
2. Bandingkan x dengan data **ke- i sampai n** (bila indeks mulai dari 0 maka hingga $n-1$)
3. Jika ada data yang sama dengan x maka cetak pesan “Ada” atau kembalikan True atau indeks dari x
4. Jika tidak ada data yang sama dengan x cetak pesan “tidak ada” atau kembalikan False atau -1

Ilustrasi Sequential Search

- Misalnya terdapat array satu dimensi sebagai berikut:

0	1	2	3	4	5	6	7	indeks
8	10	6	-2	11	7	1	100	value

- Kemudian program akan meminta data yang akan dicari, misalnya **6 (x = 6)**.
- Iterasi:
 - $6 = 8$ (tidak!)
 - $6 = 10$ (tidak!)
 - $6 = 6$ (Ya!) => output: “Ada” atau True atau index ke-2
- Jika sampai data terakhir tidak ditemukan data yang sama maka output: “Tidak ada” atau False atau -1.

Best & Worst Case

- **Best case:** jika data yang dicari terletak di depan sehingga waktu yang dibutuhkan minimal $\rightarrow O(1)$
- **Average case:** jika data yang dicari terletak di tengah-tengah $\rightarrow O(n/2)$
- **Worst case:** jika data yang dicari terletak di akhir atau tidak ada sehingga waktu yang dibutuhkan maksimal $\rightarrow O(n)$

- Contoh:

5	6	9	2	8	1	7	4
---	---	---	---	---	---	---	---

DATA

Best case ketika $x = 5$ **Average case** ketika $x=2,8$

Worst case ketika $x = 4$

* x = key/data yang dicari

Latihan

- Buatlah *flowchart* dari algoritme *Sequential Search*!

Contoh

- <http://video.franklin.edu/Franklin/Math/170/common/mod01/linearSearchAlg.html>

2. Binary Search

- Teknik pencarian = data dibagi menjadi dua bagian untuk setiap kali proses pencarian.
- Data awal harus dalam kondisi terurut, sehingga harus dilakukan proses sorting terlebih dahulu untuk data awal.
- Mencari posisi tengah:

$$\text{Posisi tengah} = (\text{posisi awal} + \text{posisi akhir}) / 2$$

Algoritme Binary Search

1. Data diambil dari posisi awal 1 dan posisi akhir N (atau posisi awal 0 dan posisi akhir N-1, tergantung indeks)
2. Kemudian cari posisi data tengah dengan rumus: **(posisi awal + posisi akhir) / 2**
3. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar?
4. Jika data sama, berarti ketemu.
5. Jika lebih besar, maka ulangi langkah 2 dengan posisi awal adalah **posisi tengah + 1**
6. Jika lebih kecil, maka ulangi langkah 2 dengan posisi akhir adalah **posisi tengah - 1**

Ilustrasi

Contoh Data:

Misalnya data yang dicari **23 (X = 23)**

Iterasi 1

0	1	2	3	4	5	6	7	8
3	9	11	12	15	17	23	31	35
A			B			C		

Karena $23 > 15$ (data tengah), maka: awal = tengah + 1

Iterasi 2

0	1	2	3	4	5	6	7	8
3	9	11	12	15	17	23	31	35
A				B			C	

**X = B (sama dengan data tengah). Output = “Ada” atau True
atau indeks 6**

Best & Worst Case

- **Best case:** jika data yang dicari terletak di posisi tengah $\rightarrow O(1)$
- **Average case:** jika data ada di sebelah kanan/kiri hingga beberapa kali pembagian $\rightarrow O(\log n)$
- **Worst case:** jika data yang dicari tidak ditemukan $\rightarrow O(\log n)$

- Contoh:

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

DATA

Bestcase ketika $x = 5$ ($T(n)=1$)

Worstcase ketika $x = 25$ ($T(n) = 5$ atau $n/2$)

* x = key/data yang dicari

Latihan

- Buatlah *flowchart* dari algoritme *binary search*!

Contoh

- <http://video.franklin.edu/Franklin/Math/170/common/mod01/binarySearchAlg.html>