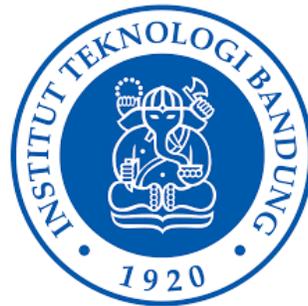


# Algoritma Decrease and Conquer

(Bagian 2)

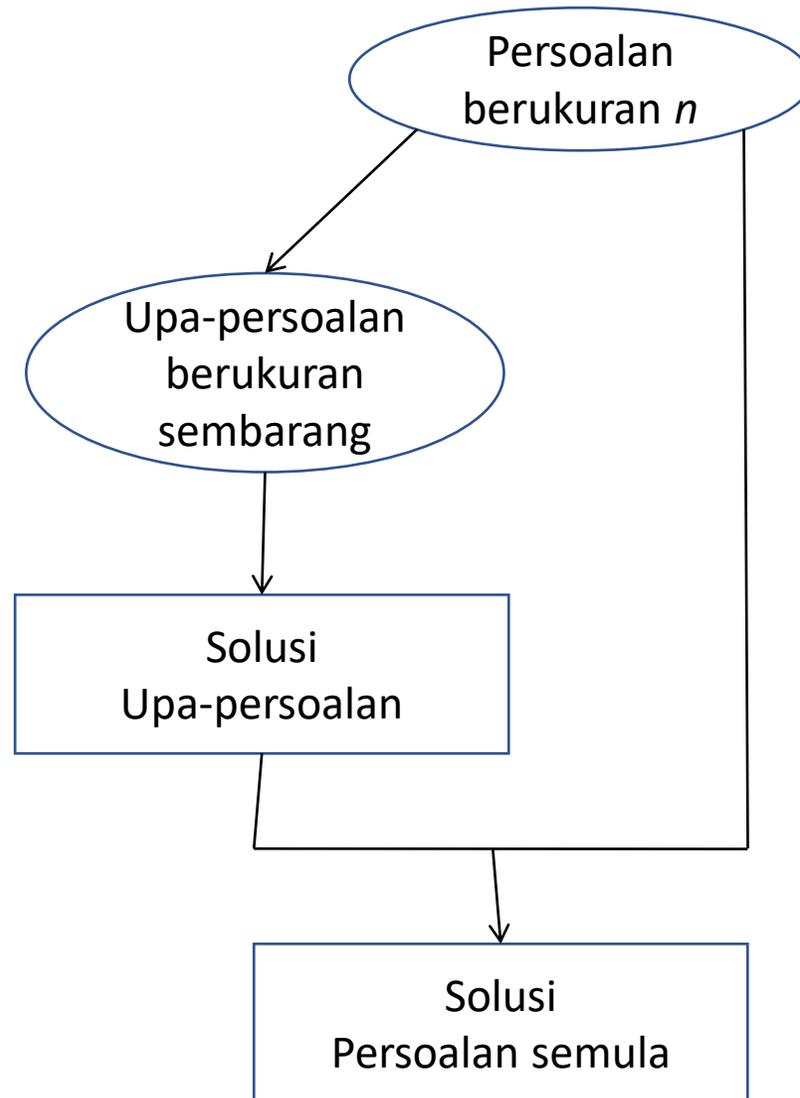
Bahan Kuliah IF2211 Strategi Algoritma

Oleh: Rinaldi Munir



Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika ITB  
2024

# *Decrease by a Variable Size*

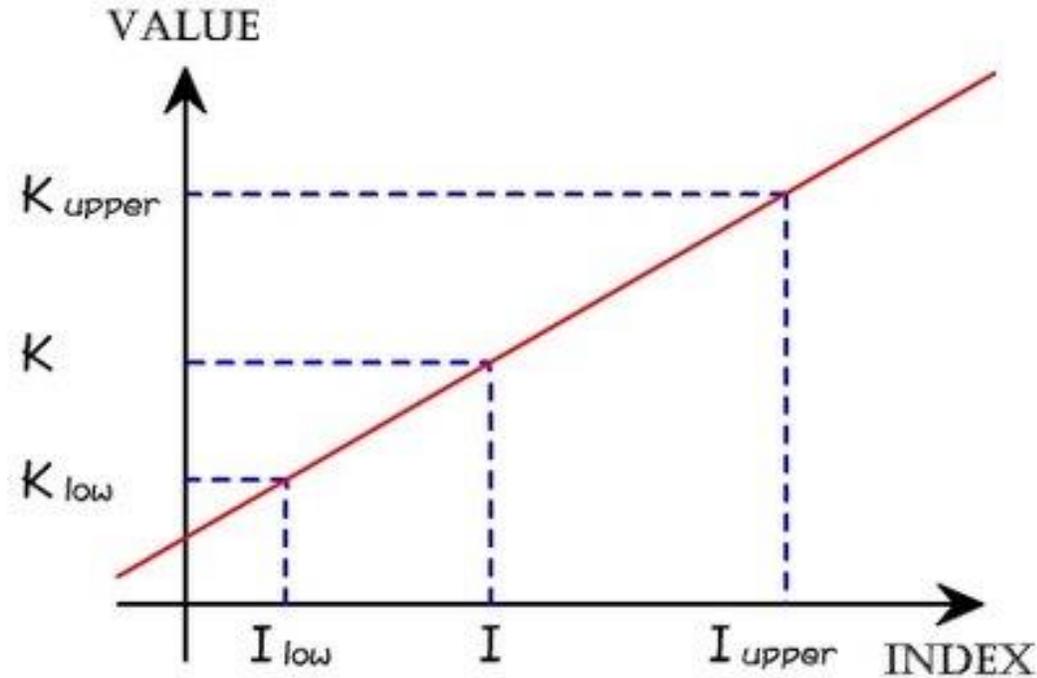


Contoh persoalan:

1. Interpolation search
2. Mencari nilai median

## ***7. Interpolation Search***

- Algoritma pencarian ini mirip dengan pencarian kata di dalam kamus atau di dalam ensiklopedi dengan cara memperkiraan letak kata tersebut di di dalam kamus
- Semua entri di dalam kamus sudah terurut menaik (dari A sampai Z).
- Memperkirakan letak kata di dalam larik dilakukan dengan teknik interpolasi.
- Kondisi awal:
  - larik A sudah terurut menaik
  - $K$  adalah nilai yang dicari



Perbandingan:

$$\frac{K - K_{low}}{K_{upper} - K_{low}} = \frac{I - I_{low}}{I_{upper} - I_{low}}$$

Perkiraan posisi  $K$  di dalam larik:

$$I = I_{low} + (I_{upper} - I_{low}) \times \frac{K - K_{low}}{K_{upper} - K_{low}}$$

$I_{low}$  adalah indeks ujung kiri larik

$I_{upper}$  adalah indeks ujung kanan larik

$K_{low}$  adalah elemen minimum di dalam larik (pada indeks  $I_{low}$ )

$K_{upper}$  adalah elemen maksimum di dalam larik (pada indeks  $I_{upper}$ )

- Algoritma *interpolation search* sama dengan algoritma *binary search*, hanya mengganti

$$mid \leftarrow (i + j) \text{ div } 2$$

dengan

$$mid \leftarrow i + (j - i) * (K - A(i)) / (A(j) - A(i))$$

sesuai dengan rumus perkiraan posisi K di dalam larik:

$$I = I_{low} + (I_{upper} - I_{low}) \times \frac{K - K_{low}}{K_{upper} - K_{low}}$$

**procedure** *Interpolationsearch*(**input**  $A : \text{LarikInteger}$ ,  $i, j : \text{integer}$ ;  $K : \text{integer}$ ; **output**  $idx : \text{integer}$ )

{ Mencari elemen bernilai  $K$  di dalam larik  $A[i..j]$  dengan *interpolation search*.

Masukan: larik  $A$  sudah terurut menaik,  $K$  sudah terdefinisi nilainya

Luaran: indek lariks sedemikian sehingga  $A[idx] = K$

}

**Deklarasi**

$mid : \text{integer}$

**Algoritma:**

**if**  $i > j$  **then**      { ukuran larik sudah 0 }

$idx \leftarrow -1$       {  $K$  tidak ditemukan }

**else**

$mid \leftarrow i + (j - i) * (K - A(i)) / (A(j) - A(i))$

**if**  $A(mid) = K$  **then**      {  $K$  ditemukan }

$idx \leftarrow mid$       { indeks elemen larik yang bernilai =  $K$  }

**else**

**if**  $A(mid) > K$  **then**

*Interpolationsearch*( $A, i, mid - 1, K, idx$ )      { cari di upalarik kiri, di dalam larik  $A[i..mid]$  }

**else**

*Interpolationsearch*( $A, mid + 1, j, K, idx$ )      { cari di upalarik kanan, di dalam larik  $A[mid+1..j]$  }

**endif**

**endif**

**endif**

- Kompleksitas algoritma *interpolation search*:
  - Kasus terburuk:  $O(n)$ , untuk sembarang distribusi data
  - Kasus terbaik:  $O(\log \log n)$ , jika data di dalam larik terdistribusi *uniform*

## 8. Mencari median dan *selection problem*.

- *Selection problem*: mencari elemen terkecil ke- $k$  di dalam sebuah senarai beranggotan  $n$  elemen.
- Jika  $k = 1 \rightarrow$  elemen paling kecil (minimum)
- Jika  $k = n \rightarrow$  elemen paling besar (maksimum)
- Jika  $k = \lceil n/2 \rceil \rightarrow$  elemen median

Bagaimana mencari median dari senarai yang tidak terurut namun tidak perlu mengurutkan senarai terlebih dahulu?

## Algoritmanya:

1. Lakukan partisi pada senarai seperti proses partisi pada algoritma *Quick Sort* (varian 2). Partisi menghasilkan setengah elemen senarai lebih kecil atau sama dengan *pivot*  $p$  dan setengah bagian lagi lebih besar dari *pivot*  $p$ .

$$\underbrace{a_{i_1} \cdots a_{i_{s-1}}}_{\leq p} \quad p \quad \underbrace{a_{i_{s+1}} \cdots a_{i_n}}_{\geq p}$$

2. Misalkan  $s$  adalah posisi pem-partisian.

Jika  $s = \lceil n/2 \rceil$ , maka pivot  $p$  adalah nilai median yang dicari

Jika  $s > \lceil n/2 \rceil$ , maka median terdapat pada setengah bagian kiri

Jika  $s < \lceil n/2 \rceil$ , maka median terdapat pada setengah bagian kanan

**Contoh 4:** Temukan median dari 4, 1, 10, 9, 7, 12, 8, 2, 15.

Pada contoh ini,  $k = \lceil 9/2 \rceil = 5$ , sehingga persoalannya adalah mencari elemen terkecil ke-5 di dalam senarai.

Partisi senarai dengan memilih elemen pertama sebagai *pivot*:

4 1 10 9 7 12 8 2 15 (indeks larik dari 1 sampai 9)

Hasil partisi:

2 1 **4** 9 7 12 8 10 15

Karena  $s = 3 < 5$ , kita memproses setengah bagian kanan:

**9** 7 12 8 10 15

8 7 **9** 12 10 15

Karena  $s = 6 > 5$ , kita memproses setengah bagian kiri:

8 7

7 8

Sekarang  $s = k = 5 \rightarrow$  stop. Jadi median = 8

- Kompleksitas algoritma:

$$T(n) = \begin{cases} a & , n = 1 \\ T(n/2) + cn & , n > 1 \end{cases}$$

- Solusi dari relasi rekurens tersebut adalah (dengan menggunakan Teorema Master):

$$T(n) = T(n/2) + cn = \dots = O(n)$$

# Soal Latihan

## 1. (UTS 2020)

Lengkapi tabel berikut ini sesuai petunjuk di tiap soal.

(a) Isikan perbandingan antara ketiga teknik dalam tabel berikut ini.

(9)

Aspek	Binary Search	Interpolation Search	Pencarian Median (Selection Problem dengan $k = \lfloor n/2 \rfloor$ )
Decrease by :			
Larik harus terurut (Ya/ Tidak)			
Kompleksitas Algoritma (Big O):			

- (b) Terdapat sebuah larik unik A sebagai berikut:  $A = [3, 14, 27, 31, 39, 42, 55, 70, 74, 81, 85, 93, 98]$ . Carilah indeks di mana nilai  $K = 85$  berada (indeks larik dimulai dari indeks 1), dengan pendekatan Binary Search dan Interpolation Search. Jika tidak ditemukan bilangan tersebut pada larik, pencarian menghasilkan -1. Tuliskan proses pencarian dengan melengkapi tabel berikut ini (penentuan nilai mid dan iterasi untuk tiap jenis pencarian). (11)

	Binary Search			Interpolation Search		
Formula pencarian indeks mid:	mid = ...			mid = ...		
Iterasi	Indeks awal	Indeks akhir	Indeks mid	Indeks awal	Indeks akhir	Indeks mid
1	1	13	...	1	13	...
2	...	...	...	...	...	...
Dst...	...	...	...	...	...	...
Indeks Akhir hasil pencarian:	...			...		
Jumlah Iterasi	...			...		

## 2. (UTS 2019)

Terdapat sebuah matriks  $A$  berukuran  $n \times n$ , yang sudah terurut menaik elemen-elemennya, sedemikian sehingga  $A[i][j] < A[i][j']$  untuk  $j < j'$ ; dan  $A[i][j] < A[i'][j]$  untuk  $i < i'$ . Persoalan yang akan diselesaikan adalah menentukan apakah sebuah elemen  $x$  ada pada matriks tersebut. Gunakan pendekatan Decrease and Conquer untuk menyelesaikan persoalan tersebut.

- (a) Tuliskan langkah-langkah pendekatan yang anda usulkan, dan tuliskan apakah pendekatan tersebut termasuk *decrease by a constant*, *decrease by a constant factor*, atau *decrease by variable size*. Tentukan juga kompleksitas pendekatan usulan anda dalam notasi Big O. **(Nilai 10)**
- (b) Terapkan pendekatan usulan anda (langkah per langkah) untuk mencari apakah  $x = 29$  terdapat pada matriks berikut ini, dan hasilkan posisi ditemukannya elemen tersebut. **(Nilai 6)**

$$\begin{bmatrix} 10 & 20 & 30 & 40 \\ 15 & 25 & 35 & 45 \\ 27 & 29 & 37 & 48 \\ 32 & 33 & 39 & 50 \end{bmatrix}$$

### 3. (UTS 2018)

*(Selection problem)* Diberikan larik (*array*) sebagai berikut:

13, 9, 18, 6, 8, 11, 15, 7, 12

Perlihatkan proses mencari elemen terbesar ke-5 dengan algoritma *decrease and conquer* dan memanfaatkan algoritma partisi dari algoritma *Quicksort* varian kedua. *Pivot* yang diambil selalu elemen pertama larik. (Nilai: 12)

TAMAT