Soal Latihan
Algoritma Divide and Conquer 1

IF2211 Strategi Algoritma

By: Rinaldi M

UTS 2022

Diberikan sebuah larik yang berisi elemen biner (0 atau 1). Elemen-elemen larik sudah **terurut** menaik (dari kecil ke besar). Kita akan menghitung jumlah bit 1 di dalam larik tersebut.

Contoh: A = [0, 0, 0, 1, 1, 1, 1, 1], jumlah bit 1 adalah 5.

Jika diselesaikan dengan algoritma *divide and conquer*, jelaskan caranya atau langkah-langkahnya, lalu tentukan kompleksitas waktu asimptotiknya dalam notasi O besar.

Jawaban:

HitungBit(A, n, count)

(i) Basis: Jika n = 1, maka periksa apakah A[n] = 1. Jika A[n] = 1, maka count = 1, else count = 0

(ii) Rekurens: Jika ukuran larik > 1, maka

- Bagidua larik pada posisi pertengahan, A1 dan A2, masing-masing larik berukuran n/2
- **jika** pada upalarik kiri, A1, elemen terakhirnya 0, maka dipastikan seluruh elemen larik pada bagian tersebut adalah 0 (karena larik sudah terurut menaik), jadi count1 = 0.

  else HitungBit1(A1, n/2, count1)
- **jika** elemen pertama pada upalarik kanan, A2, adalah 1, maka dipastikan seluruh elemen pada bagian larik tersebut adalah 1. Jadi, count2 = ujung kanan – ujung kiri + 1.

  else HitungBit(A2, n/2, count2)
- count = coun1 + count1

Kompleksitas waktu algoritma, dihitung dari jumlah operasi perbandingan elemen, adalah:

$$T(n) = 1, \text{ jika } n = 1$$

$$= T(n/2) + 2, \text{ jika } n > 1$$

Menurut teorema master, $a = 1$, $b = 2$, $d = 0$, sehingga $a = b^d$ (case 2), jadi $T(n) = O(n^0 \log n) = O(\log n)$

UTS 2020

Sebuah larik A berisi deretan bilangan integer yang tidak terurut. Tuliskan algoritma dengan pendekatan **Divide and Conquer** dalam bentuk rekursif untuk mencari banyaknya elemen yang memiliki rentang nilai di antara (dan termasuk) *min* dan *maks*. Sebagai contoh, jika A = 10, 29, 89, 50, 34, 91, 39, 66, 20 dengan nilai yang dicari dalam rentang antara *min*=20 dan *maks*=40, maka banyak elemen dengan rentang tersebut adalah 4 (empat).

Tuliskan kompleksitas waktu algoritma tersebut berdasarkan jumlah operasi perbandingan elemen.

Jawaban:

```
function CountRange(A[i..j], min, maks)
if  i = j then              { larik hanya berisi satu elemen }
    if  A[i] ≥ min AND A[i] ≤ maks then
        return 1
    else
        return 0
    endif
else
    mid = (i + j) div 2
    return  CountRange (A, i, mid, min, maks) + CountRange (A, mid+1, j, min, maks)
endif
```

Kompleksitas waktu:

$$T(n) = \begin{cases} 2 & , n = 1 \\ 2T(\frac{n}{2}) & , n > 1 \end{cases}$$

Relasi rekurens ini tidak dapat diselesaikan dengan Teorema Master (mengapa?).
Selesaikan secara iteratif:

Misalkan $n = 2^k$

$T(n) = 2T(n/2) = 2(2T(n/4) = 4T(n/4) = 4(2T(n/8) = 8T(n/8) = \ldots = 2^k T(1)$

$= 2^k (2) = 2n = O(n)$

Cara lainnya:

$$T(n) = \begin{cases} 2 & , n = 1 \\ 2\,T(\frac{n}{2}) & , n > 1 \end{cases}$$

$$T(n) = 2\,T\left(\frac{n}{2}\right)$$

Missal $n = 2^k$

$$T(n) = 2T\left(\frac{2^k}{2}\right)$$

$$= 2\ T(2^{k-1}) = \ 2^2\ T(2^{k-2}) = \ 2^3\ T(2^{k-3})$$

$$= 2^i\ T\left(2^{k-i}\right)$$

Saat i = k, maka

$$T(n) = 2^k\ T(2^0) = n\ T(1) = 2\,n = O(n)$$