2023-2024-2 | Perancangan dan Analisis Algoritma | Tugas 2

Mohammad Hanif Furqan Aufa Putra | 5025221161 | PAA - A

1. Substitution Method

Buktikan menggunakan *Substitution Method* bahwa $T(n) = T(n-1) + n$ adalah $O(n^2)$!

2. Iteration Method

Tentukan *upper bound* dari $T(n) = 2T(n-1) + 1$ menggunakan *Iteration Method*!

3. Master Method

Tentukan *tight asymptotic bounds* (Θ) untuk fungsi *reccurences* berikut ini:

1. $T(n) = 2T(n/4) + 1$
2. $T(n) = 2T(n/4) + \sqrt{n}$
3. $T(n) = 2T(n/4) + n$
4. $T(n) = 2T(n/4) + n^2$

Jawaban:

1. Substitution Method
    a. Buat asumsi T(n) = O(n^2).
    b. Buat asumsi T(k) ≤ ck^2 untuk k ≤ n-1.
    c. Membuktikan T(n) ≤ cn^2 dengan menggunakan induksi.
    T(n) = T(n-1) + n
    ≤ c(n-1)^2 + n
    = c(n^2 - 2n + 1) + n
    = cn^2 - 2cn + c + n
    = cn^2 - (2cn - c - n)
    ≤ cn^2 jika c ≥ 1 dan n ≥ 1.

2. Iteration Method

   $T(n) = 2T(n - 1) + 1$

   $T(n) = 2(2T(n - 2) + 1) + 1 = 4T(n - 2) + 3$

   $T(n) = 4(2T(n - 3) + 1) + 3 = 8T(n - 3) + 7$

   $T(n) = 8(2T(n - 4) + 1) + 7 = 16T(n - 4) + 15$

   $T(n) = 2^k T(n - k) + (2^k - 1)$

   Mencari nilai k di mana n - k = 1, yang berarti k = n - 1. Jadi:

   $T(n) = 2^{n-1} T(1) + 2^{n-1} - 1$

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

   Jadi, upper bound dari $T(n) = 2T(n - 1) + 1$ $adalah$ $0(2^n)$

3. Master Method

   a. $a = 2;\ b = 4;\ f(n) = 1$

      $n^{log_b a} = n^{log_4 2} = n^{1/2}$

      Karena waktu eksekusi f(n) = 1 lebih rendah daripada $n^{1/2}$, maka penyelesaian untuk masalah ini adalah: $T(n) = \theta(n^{1/2})$.

   b. $a = 2;\ b = 4;\ f(n) = \sqrt{n}$

      $n^{log_b a} = n^{log_4 2} = n^{1/2}$

      Dikarenakan $f(n) = \sqrt{n}$ mempunyai waktu eksekusi yang setara dengan $n^{1/2}$, solusinya adalah: $T(n) = \theta(n^{log_b a} log\ n) = \theta(n^{1/2} log\ n)$

   c. $a = 2;\ b = 4;\ f(n) = n$

      $n^{log_b a} = n^{log_4 2} = n^{1/2}$

      Karena f(n) = n memiliki waktu eksekusi yang lebih cepat daripada $n^{1/2}$, maka solusinya adalah: $T(n) = \theta(f(n)) = \theta(n)$

d. $a = 2;\ b = 4;\ f(n) = n^{2}$
$n^{\log_b a} = n^{\log_4 2} = n^{1/2}$

Karena $f(n) = n^{2}$ memiliki waktu eksekusi yang lebih cepat daripada $n^{1/2}$, maka solusinya adalah: $T(n) = \theta(f(n)) = \theta(n^{2})$.