



Analisa Efisiensi Algoritma Rekursif

Elsa Elvira Awal, M.Kom

Review Materi Sebelumnya

Algoritma fib(n)

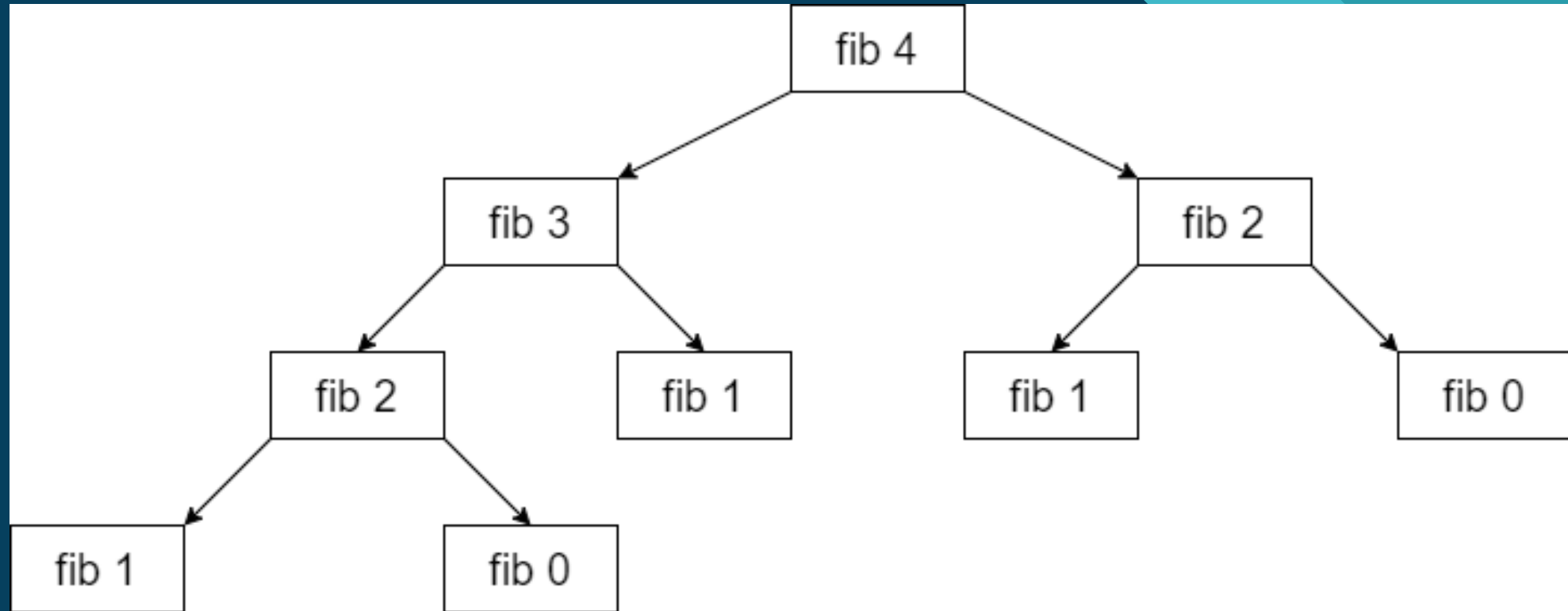
// Algoritma untuk menghitung bilangan Fibonacci ke n secara rekursif

// Input: n

// Output: bilangan Fibonacci ke n

```
if n = 0 or n = 1
    return n
else return (fib(n-1) + fib(n-2))
```

Pohon Rekursif



Pohon Rekursif

- Untuk mengeksekusi 1 node, butuh 1 satuan waktu
- Makin banyak node, waktu eksekusi makin lama
- Untuk mengeksekusi 1 node butuh 1 satuan memori
- Makin banyak node, memori yang dibutuhkan semakin besar

Analisa Efisiensi Algoritma Rekursif

Algoritma pangkat (X, n)

// Algoritma untuk menghitung nilai X^n secara rekursif

// Input: integer positif X dan n

// Output: nilai X^n

```
if n = 1
    return X
else return (X * pangkat (X, n-1))
```

Langkah-langkah Umum untuk Menganalisa Efisiensi Waktu Algoritma Rekursif

- Tentukan metric untuk ukuran input
- Identifikasi basic operation algoritma
- Tentukan apakah untuk ukuran input yang sama, banyaknya eksekusi basic operation bisa berbeda
- Tentukan persamaan rekursi yang menunjukkan berapa kali basic operation dieksekusi
- Cari rumus langsung yang menunjukkan banyaknya basic operation dieksekusi

Tentukan metric untuk ukuran input

- Sesuatu pada input yang jika membesar, maka banyaknya pemanggilan fungsi rekursif bertambah
- Pada kasus ini adalah nilai n . jika n membesar, maka banyaknya komputasi atau pemanggilan fungsi rekursi bertambah
- Untuk memahaminya coba gambar pohon rekursifnya
- Efisiensi dinyatakan sebagai fungsi dari n .

Identifikasi basic operation algoritma

- Pada algoritma rekursif merupakan salah satu operasi pada kondisi seleksi base case atau bagian recursive case
- Basic operationnya dipilih = yang dilakukan 1 kali setiap kali fungsi rekursif dipanggil
- Pada kasus X^n basic operationnya adalah $X * \text{pangkat}(X, n-1)$

Tentukan apakah untuk ukuran input yang sama, banyaknya eksekusi basic operation bisa berbeda

- Pada algoritma pangkat rekursif ini, untuk nilai n yang sama, jumlah perkalian yang dilakukan akan selalu sama. Ini karena algoritma akan melakukan satu perkalian di setiap langkah sampai mencapai $n=1$.

Pohon Rekursif

Pangkat (5, 4) = 5 * pangkat (5, 3)



625

5 * pangkat (5, 2)

5 * pangkat (5, 1)

pangkat (5, 1) = 5

Tentukan persamaan rekursi yang menunjukkan berapa kali basic operation dieksekusi

- Jika algoritma pangkat dieksekusi dengan input (X, n) maka basic operation dieksekusi satu kali.
- Namun pada saat eksekusi, algoritma tersebut juga memanggil dirinya sendiri dengan $(X, n - 1)$

Tentukan persamaan rekursi yang menunjukkan berapa kali basic operation dieksekusi

- Jika $C(n)$ menyatakan banyaknya basic operation dieksekusi untuk input berukuran n dan $C(n-1)$ menyatakan banyaknya basic operation dieksekusi untuk input berukuran $n-1$
- Hubungan $C(n)$ dan $C(n-1)$ dinyatakan dengan
- $C(n) = C(n-1) + 1$ untuk $n > 1$ (recursive case)
- $C(1) = 1$, base case

Hubungan Rekursif $C(n) = C(n-1) + 1$

- $C(n)$ menyatakan jumlah basic operation yang dieksekusi untuk menghitung X^n
- Setiap kali algoritma dipanggil dengan input n , ia memerlukan satu operasi tambahan (perkalian) dibandingkan dengan jumlah operasi yang diperlukan untuk ukuran input $n-1$
- Oleh karena itu, hubungan antara jumlah operasi untuk ukuran input n dan $n-1$ adalah $C(n) = C(n-1) + 1$
- Ini berarti bahwa untuk menghitung pangkat dengan input n , kita perlu melakukan satu operasi tambahan.

Cari rumus langsung yang menunjukkan banyaknya basic operation dieksekusi

- Untuk algoritma pangkat, jumlah operasi dasar yang dilakukan dinyatakan dengan $C(n) = C(n-1) + 1$
- Ini artinya, jumlah operasi untuk menghitung pangkat dengan ukuran n (yaitu $C(n)$) adalah jumlah operasi untuk ukuran $n-1$ ($C(n-1)$), ditambah satu operasi tambahan (perkalian yang dilakukan saat ini)
- $C(2)=C(1)+1 = 1 + 1 = 2$
- $C(3)=C(2)+1 = 2 + 1 = 3$
- $C(4)=C(3)+1 = 3 + 1 = 4$



Thank You