ANALISIS ALGORITMA REKURSIF SUBTITUSI

Pengertian Rekursif

Rekursif → Fungsi yang memanggil dirinya sendiri.

Syarat Rekursif :

- Base : kondisi penghenti
- Reccurence (rekuren) : pemanggilan fungsi kembali (memanggil dirinya sendiri).

Contoh Rekursif (Menghitung Faktorial)

```
Function F(input n: integer) →real
Kamus :
  fak : real;
Algoritma
  if n=0 then
    fak ← 1                     Base
  else
    fak ← n *F(n-1)             Rekuren
  endif
  return fak
endFunction
```

Teknik Analisis Rekursif

Teknik untuk analisis algoritma rekursif:

- Subtitution
- Characteristic Equation

General Plan

- Tentukan ukuran
- Tentukan operasi dasar
- Cek apakah terdapat best, worst atau average casenya.
- Tentukan relasi rekuren dan kondisi inisialnya.
- Selesaikan rekuren.

- Recurrence:

$$\boxed{\begin{aligned} &t_n = t_{n-1} + n \qquad n > 1 \\ &t_1 = 1 \end{aligned}}$$

- Work backward:

$$\begin{aligned} t_n &= t_{n-1} + n \\ t_{n-1} &= t_{n-2} + n - 1 \\ t_{n-2} &= t_{n-3} + n - 2 \\ &\vdots \\ t_2 &= t_1 + 2 \\ t_1 &= 1. \end{aligned}$$

Substitution (Lanjutan)

- Substitution:

$$
\begin{aligned}
t_n &= t_{n-1} + n \\
&= t_{n-2} + n - 1 + n \\
&= t_{n-3} + n - 2 + n - 1 + n \\
&\;\vdots \\
&= t_1 + 2 + \cdots + n - 2 + n - 1 + n \\
&= 1 + 2 + \cdots + n - 2 + n - 1 + n \\
&= \sum_{i=1}^{n} i = \frac{n(n+1)}{2}.
\end{aligned}
$$

Contoh Rekursif (Menghitung Faktorial)

```
Function F(input n: integer) →real
Kamus :
  fak : real;
Algoritma
  if n=0 then
    fak ← 1                    Base
  else
    fak ← n *F(n-1)            Rekuren
  endif
  return fak
endFunction
```

- Berdasarkan contoh algoritma faktorial diatas :
- Operasi dasar : perkalian
- Fungsi Faktorial F(n) berdasarkan rumus:

  $$F(n) = n * F(n-1), \quad \text{untuk } n>0$$

- Untuk kasus basis tidak ada operas perkalian (0)
- Untuk kasus rekuren, kompleksitas diukur dari jumlah perkalian (1) ditambah kompleksitas waktu untuk faktorial F(n-1). Misalkan F(n-1) diinisialisasi dengan T(n-1) maka :

  $$T(n) = 1 + T(n-1) \Rightarrow \text{recurrence rel}$$

- Kondisi inisialnya:

  $$M(0) = 0$$

- Jadi relasi rekurens :

$$T(n) = \begin{cases} 0 & , n = 0 \\ T(n-1)+1 & , n > 0 \end{cases}$$

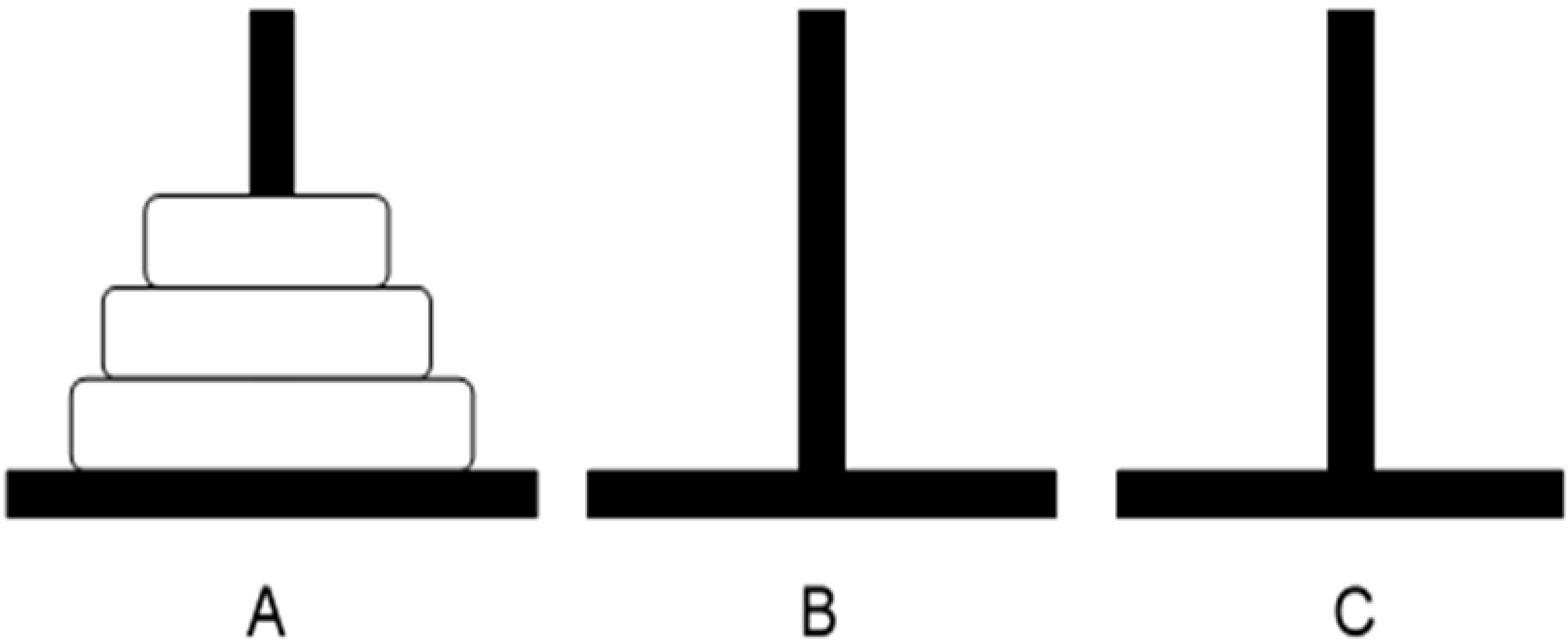
$$\begin{aligned} T(n) &= 1 + T(n-1) \\ &= 1+1+T(n-2) = 2+T(n-2) \\ &= 2+1+T(n-3) = 3+T(n-3) \\ &= \ldots.. \\ &= n + T(0) \end{aligned}$$

$\longrightarrow$ T(n-n) = 0 → menggunakan kondisi inisial

$$= n + 0$$

Jadi T(n) = n → O(n)

Towers of Hanoi (Menara Hanoi)



Solusi Rekursif untuk menara Hanoi

- Pindahkan secara rekursif (n-1) disk dari A ke B (dengan C sebagai bantunya)
- Pindahkan disk terbesar dari A ke C
- Pindahkan secara rekursif (n-1) disk dari B ke C (dengan A sebagai bantunya)

- Operasi Dasar : Memindahkan Disk
- Rekurennya :

$$T(n) = T(n-1)+1+T(n-1), \text{ untuk } n > 1$$

$$= 2T(n-1)+1, \text{ untuk } n > 1$$

$$M(1) = 1$$

- Relasi Rekuren nya :

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

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

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

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

$$= \ldots\ldots.$$

$$= \left(1 + 2 + 2^2 + \ldots\ldots. + 2^{n-2}\right) + 2^{n-1} T(1)$$

$$= 1 + 2 + 2^2 + \ldots.. + 2^{n-1}.1$$

$$= 2^n - 1$$

Latihan

```
Algortihm S(n)
{input : A positive integer n}
{Output : the sum of the first n cubes}

If n=1 return 1
Else return S(n-1) + n * n* n
```

- Operasi Dasar?
- Relasi rekuren ?
- T(n) ?