

Kasus Kompleksitas

Alfansyah Putra Raja Dinata

L0223002 – Sains Data B

I. Contoh Kasus Kompleksitas Algortima

1. Algoritma Tower of Hanoi (Rekursif)

Tower of Hanoi adalah sebuah teka teki matematika dengan menggunakan 3 disk dan 3 pilar dengan tujuan untuk memindahkan seluruh tumpukan dari batang 1 ke batang lain.

dalam pengerjaannya, terdapat kode seperti ini:

```
TOH(n, x, y, z)
{
    jika (n >= 1)
    {
        // taruh (n-1) disk ke z dengan menggunakan y
        TOH((n-1), x, z, y)

        // pindahkan disk yang lebih besar ke tempat yang tepat
        pindah:x-->y

        // taruh (n-1) disk di tempat yang tepat
        TOH((n-1), z, y, x)
    }
}
```

Dengan algoritma tersebut, kita bisa melihat persamaan rekursif seperti ini:

$$T(n) = 2T(n - 1) + 1 \quad \text{untuk persamaan 1}$$

Lalu dapat di substitusi balik dengan :

$$T(n - 1) = 2T(n - 2) + 1 \quad \text{untuk persamaan 2}$$

$$T(n - 2) = 2T(n - 3) + 1 \quad \text{untuk persamaan 3}$$

Kita bisa memasukkan nilai $T(n-2)$ ke dalam persamaan 2 dengan bantuan persamaan 3 :

$$T(n - 1) = 2(2T(n - 3) + 1) + 1 \quad \text{untuk persamaan 4}$$

Terakhir kita bisa mencoba untuk melakukan perhitungan dengan memasukkan nilai $T(n-1)$ di persamaan 1 dengan bantuan persamaan 4 :

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

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

Yang hasil setelah digeneralisasi nya :

$$T(n) = 2^kT(n - k) + 2^{(k-1)} + 2^{(k-2)} + \dots + 2^2 + 2^1 + 1$$

Dimana Kondisi dasar nya adalah

$$T(1) = 1$$

$$n - k = 1$$

$$k = n - 1$$

$$T(n) = 2^kT(n - k) + 2^{(k-1)} + 2^{(k-2)} + \dots + 2^2 + 2^1 + 1$$

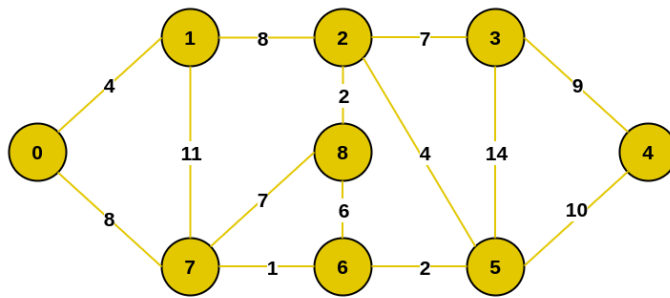
Karena ini deret geometri maka hasilnya adalah $2^n - 1$

jadi:

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

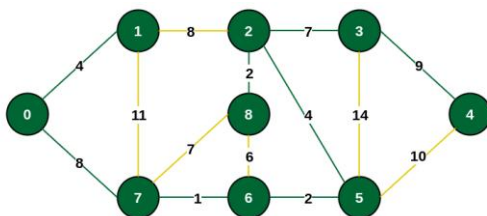
2. Algoritma Minimum Spanning Tree Prsim

Penggunaan Algoritma untuk Minimum Spanning Tree Prsim adalah dengan melakukan pencarian terhadap *vertex* dan membandingkan pada bagian mana yang lebih ringan.

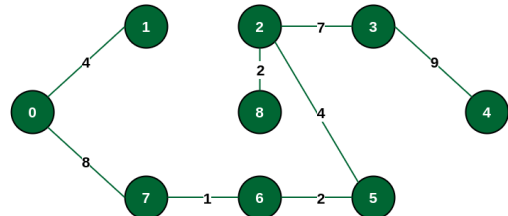


Example of a Graph

Dia akan mencari bagian bagian mana yang lebih ringan sampai akhirnya *path* telah ditemukan.

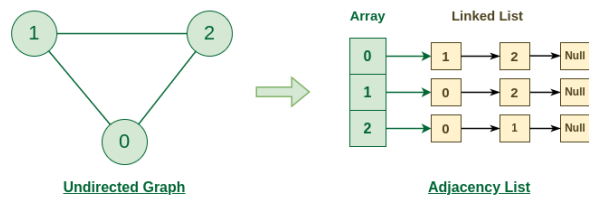


Minimum weighted edge from MST to other vertices is 3-4 with weight 9



The final structure of MST

Algoritma MST ini menggunakan kedekatan *list*,



Undirected Graph

Adjacency List

Graph Representation of Undirected graph to Adjacency List

Yang mana kompleksitas waktu nya adalah $T(n) = O(V^2)$ karena untuk mencari tiap cabang menggunakan sistem membandingkan 2 vertex dengan skema n yang berpangkat .

II. Contoh Soal

1. Tunjukkan bahwa $T(n) = 2n^2 + 6n + 1 = O(n^2)$ dimana $C = \text{konstan}$ dan $n \geq n_0$

Gunakan 4 untuk mengisi C :

$$2n^2 + 6n + 1 \leq 4n^2$$

$$6n + 1 \leq 4n^2 - 2n^2$$

$$6n + 1 \leq 2n^2$$

$$\rightarrow 6/n + 1/n^2 \leq 2$$

Kita bisa mencoba untuk tes untuk mencari n_0 :

$$N=2$$

$$6/2 + 1 / 2^2 \leq 2$$

$$3.25 \leq 2 \rightarrow \text{salah}$$

$$N = 4$$

$$6/4 + 1 / 4^2 \leq 2$$

$$1.5626 \leq 2 \rightarrow \text{benar}$$

2. Tunjukkan bahwa $T(n) = 3n + 2 = O(n) \leq 3n + 2n = 5n$ untuk semua $n \geq 1$ ($C = 5$ dan $n_0 = 1$)

karena $3n + 2 = O(n)$

$$\rightarrow T(n) \quad 3n + 2 \leq Cn$$

$$\rightarrow 2 \leq Cn$$

$$\rightarrow 2 \leq (C - 3) n$$

A) $C = 4$

$$- \quad 2 \leq (4 - 3) n$$

$$- \quad 2 \leq n$$

B) $C = 7$

$$- \quad 2 \leq (7 - 3) n$$

$$- \quad 2 \leq 4n : 2 = 1 \leq 2n$$