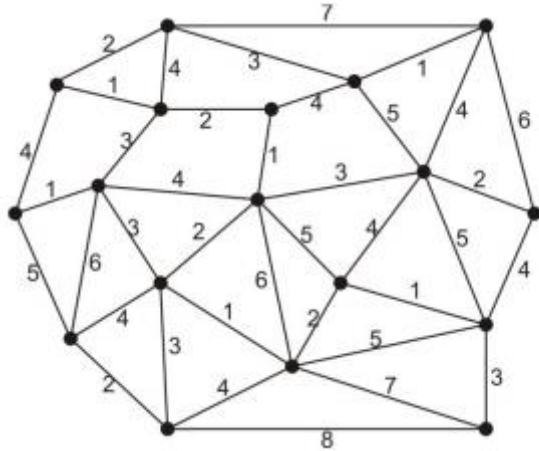


Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Rekursif dan Relasi Rekurens, Pohon, dan Kompleksitas Algoritma
 Dosen: Rinaldi Munir, Harlili
 Rabu, 2 Desember 2015
 Waktu: 75 menit

1. Graf di bawah ini merepresentasikan jarak antar 18 kota dalam suatu pulau. Tentukan total jarak minimum yang menghubungkan semua kota tersebut menggunakan pohon merentang minimum. Gunakan algoritma Prim! Gambarkan pohon merentang minimum yang dimaksudkan!



2. Misalkan terdapat sebuah pohon P yang merupakan pohon 3-ary teratur dengan tinggi 4.
- Tentukanlah jumlah daun P dan jumlah simpul P !
 - Misalkan dilakukan penghapusan semua anak (beserta keturunannya) dari salah satu simpul pada aras 2. Tentukanlah kembali jumlah daun P dan jumlah simpul P !
3. Misalkan terdapat string: “RAJA PADJAJARAN”
- Gambarkan pohon Huffman dengan terlebih dulu menghitung frekuensi kemunculan tiap karakternya (termasuk spasi)
 - Tentukan kode huffman untuk masing-masing karakter dalam bentuk tabel lalu hitung panjang rangkaian bit yang dihasilkan jika string diatas diubah menjadi kode huffman yang telah dibuat
 - Tentukan kata yang terbentuk dari rangkaian bit 10010001 dengan proses decoding menggunakan kode huffman diatas (jika tidak ada cukup tulis “tidak ada”).
4. Misalkan terdapat operasi biner: penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan perpangkatan (\uparrow). Diketahui suatu ekspresi dalam bentuk postfix
- $$6\ 2\ \uparrow\ 9\ /\ 7\ 2\ 3\ \uparrow\ +\ 5\ /\ +$$
- Tentukan pohon ekspresi dan nilai dari ekspresi postfix tersebut!
 - Tentukan bentuk prefix dan infiks dari ekspresi postfix tersebut!
5. Tentukan notasi O , Ω dan Θ untuk $T(n) = 2n^2 + 7n \log n + 1!$
6. Tentukan solusi relasi rekurens berikut:

$$a_n = \frac{a_{n-2}}{4} \text{ untuk } n \geq 2, a_0 = 1, a_1 = 0$$

7. Diketahui prosedur `foo` dalam notasi *Pascal-like* seperti dibawah:

```
function foo(n: integer) : integer
var i,j,k: integer

Algoritma
  i:=n
  while(i>0) do
    j:=1
    while(j<n) do
      k:=0
      while(k<n) do
        sum:=sum + (i + j + k)
        k:=k +2
      end while
      j:= j * 2,
    end while
    i:=i/2
  end while
  return sum
```

Tentukan kompleksitas algoritmanya jika ditinjau berdasarkan *assignment* nilai pada instruksi `sum:=sum + (i + j + k)!`