

Solusi Kuis ke-4 IF2091 Struktur Diskrit (3 SKS): Pohon dan Kompleksitas Algoritma  
 Dosen: Rinaldi Munir, Harlili  
 Senin, 30 November 2011  
 Waktu: 55 menit

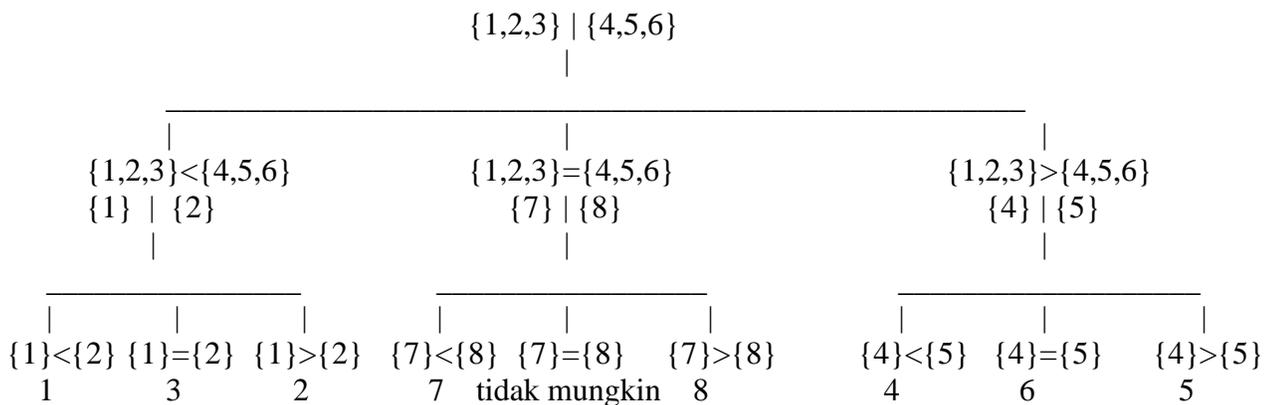
1. Diberikan 7 buah koin, semua berat koin sama terkecuali satu koin *defect* dengan berat yang kurang dari berat koin lainnya. Tentukan jumlah penimbangan minimum sehingga kita bisa menentukan koin mana yang paling ringan diantara semua koin. Selain itu tunjukkan dengan membuat sebuah pohon keputusan di mana simpul pada pohon ditulis sebagai  $(\{A\}, \{B\})$  di mana  $\{A\}$  dan  $\{B\}$  merupakan himpunan nomor koin yang ditimbang (tiap koin ditandai dengan nomor 1,2,3,4,5,6,7). Himpunan koin  $\{A\}$  ditimbang dibagian kiri timbangan dan himpunan koin  $\{B\}$  ditimbang dibagian kanan timbangan.

Jawaban:

Paling optimal ada 2 kali penimbangan yang harus dilakukan, perlu diperhatikan bahwa tiap node punya tiga anak/child yang merupakan cabang dari tiap kasus, yaitu :

1.  $\{A\}$  lebih ringan dari  $\{B\}$ .
2.  $\{A\}$  sama berat dengan  $\{B\}$ , dan
3.  $\{A\}$  lebih berat dari  $\{B\}$ .

Pohon solusi :



Bisa dilihat pohon keputusan di atas menyusun semua kemungkinan penimbangan, bisa dibuktikan bahwa dibutuhkan minimum 2 kali penimbangan untuk mengetahui koin yang *defect*

2. Sebuah pohon 5-ary tingginya 6. Tanpa menggambarkan pohonnya, hitung jumlah maksimum seluruh simpul dan daun di dalam pohon tersebut.

Jawaban:

Jumlah simpul pada ketinggian  $i$  yaitu  $5^i$ .

Jadi jumlah maksimum seluruh simpul pada pohon tersebut yaitu :

$$5^0 + 5^1 + 5^2 + 5^3 + 5^4 + 5^5 + 5^6 = \frac{5^0(5^7 - 1)}{5 - 1} = 19531$$

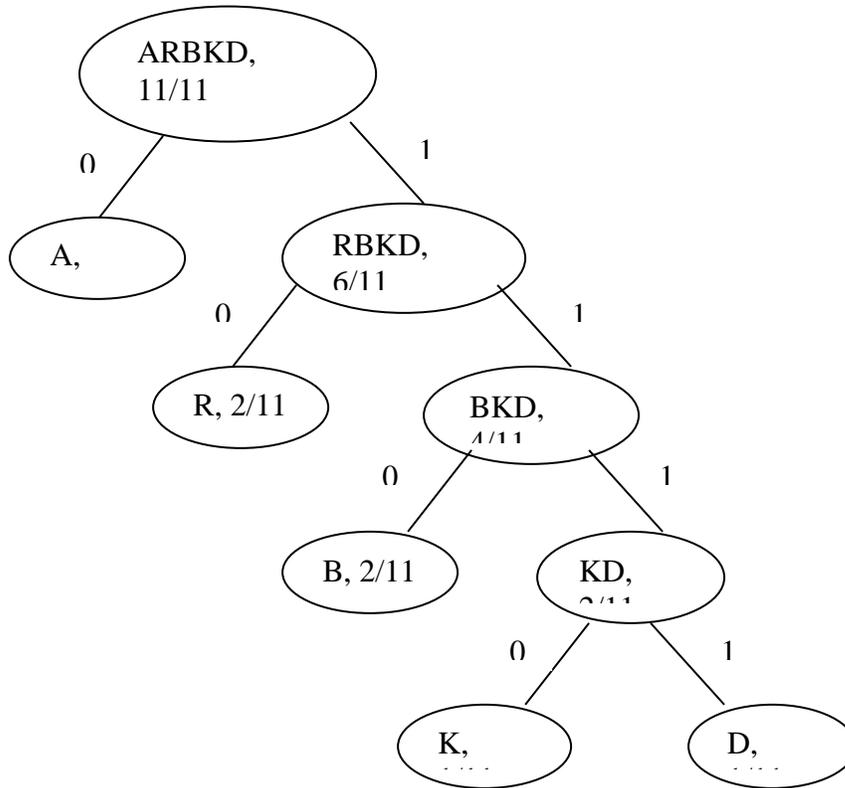
Jumlah daun pada pohon tersebut :  $5^6 = 15625$

3. Susunlah representasi bit *Huffman Encoding* dari *string* berikut dengan bantuan pohon Huffman:  
 "ABRAKADABRA"

Jawaban:

Jumlah huruf A = 5      Peluang(A) = 5/11  
 Jumlah huruf R = 2      Peluang(R) = 2/11  
 Jumlah huruf B = 2      Peluang(B) = 2/11  
 Jumlah huruf K = 1      Peluang(K) = 1/11  
 Jumlah huruf D = 1      Peluang(D) = 1/11

Cari representasi bit menggunakan pohon Huffman :



Maka representasi bit dari masing-masing huruf :

A = 0  
 B = 110  
 K = 1110  
 D = 1111  
 R = 10

Maka representasi string "ABRAKADABRA" adalah :

01101001110011110110100

4. Berikan estimasi Big-O untuk  $T(n) = (n + 8)\log(n^2 + 2) + 7n^2$

**Jawaban:**

Misalkan,  $f(n) = n+5$ ,  $g(n) = \log(n^2+2)$ , dan  $h(n) = 3n^2$

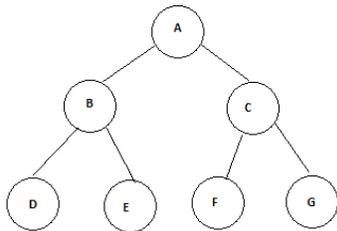
Untuk  $f(n) = n + 5$ ,  
 $f(n) = O(n)$

Untuk  $g(n) = \log(n^2+2)$ ,  
 $\log(n^2+1) \leq \log(2n^2) = \log 2 + \log n^2 = \log 2 + 2 \log n \leq \log n$   
 Maka  $g(n) = O(\log n)$

Untuk  $h(n) = 3n^2$ ,  
 $h(n) = O(n^2)$

Notasi Big-O untuk  $T(n) = f(n)g(n) + h(n) = \max(n \log n, n^2) = O(n^2)$

5. Gambar dibawah ini merupakan contoh pohon biner penuh dengan kedalaman 2.



Misalkan simpul/node pada pohon direpresentasikan dalam struktur data berikut (dalam bahasa Java):

```
public class Node {
    private int _index;
    private Boolean _isLeaf;
    private Node[] _child;
    private Node _parent;
}
```

Misalkan terdapat suatu pohon biner penuh yang memiliki kedalaman  $n$  dengan root bernama *learningModel*. Apabila dilakukan prosedur `traverse(learningModel)`, tentukan  $T(n)$  dan  $O(n)$  jika melihat banyaknya proses `System.out.println("Kuis Strukdis");`

```
public void traverse(Node node) {
    if (!node.getIsLeaf()) {
        System.out.println("Kuis Strukdis");
        Node[] childs = node.getChild();
        for (int i=0; i < childs.length; ++i) {
            traverse(childs[i]);
        }
    }
}
```

**Jawaban:**

Prosedur `traverse(Node node)` akan melakukan traversal pohon biner dan akan melakukan print "Kuis Strukdis" apabila simpul yang sedang di cek bukan merupakan daun.

Karena kedalaman pohon adalah  $n$ , maka `traverse(learningModel)` akan melakukan proses

`System.out.println("Kuis Strukdis")` sejumlah simpul yang ada pada pohon tersebut tanpa simpul daun dihitung.

$$T(n) = 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-1} = 2^n - 1$$

Dalam Big-O :  $O(2^n)$