

22. Tinggi sebuah binary tree adalah 0 jika hanya memiliki satu node yaitu root saja. Maksimum dari jumlah leaf pada binary tree yang berisikan 66 node adalah .....

B. 18 C. 29 D. 33

Perhatikan pada perfect binary tree dengan ketinggian 5 memiliki max\_node =  $2^{(K+1)} - 1 = 63 \text{ nodes}$ 

Jumlah leaf node pada perfect binary tree 2/5 = 32 nodes

Penambahan 3 node pada indeks selanjutnya akan menyebabkan perfect binary tree menjadi complete binary tree dengan jumlah leaf sebanyak 33 nodes -> pilihan (b), (c) salah

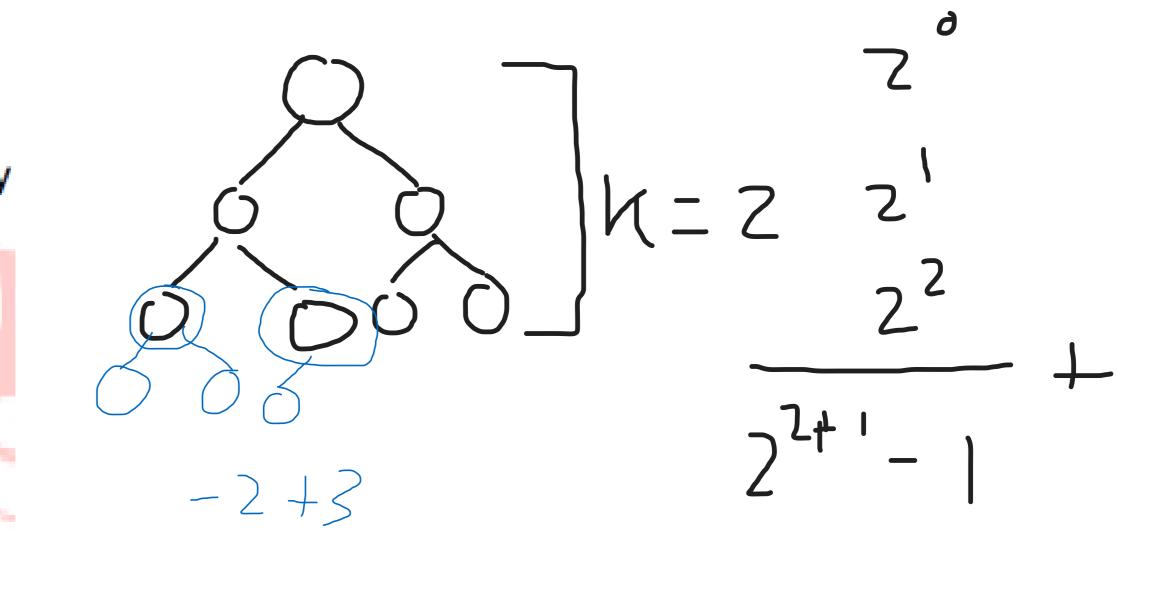
Jumlah maksimum node dalam binary tree dengan tinggi *k* adalah 2<sup>k+1</sup> -1.

Sebuah full binary tree adalah binary tree yang semua node-nya memiliki 0 atau 2 anak (children).

 Sebuah complete binary tree dengan tinggi k adalah binary tree yang miliki jumlah maximum nodes di levels 0 sampai k – 1 (semua level terisi kecuali pada level terakhir, yang terisi dari sisi kiri hingga kanan dan tidak memiliki *missing* nodes).

Sebuah perfect binary tree dengan tinggi k adalah sebuah binary tree yang memiliki  $2^{k+1}$  - 1 nodes. Source: Weiss, Mark Allen. 2014. Data structures and algorithm analysis in C++ / Mark

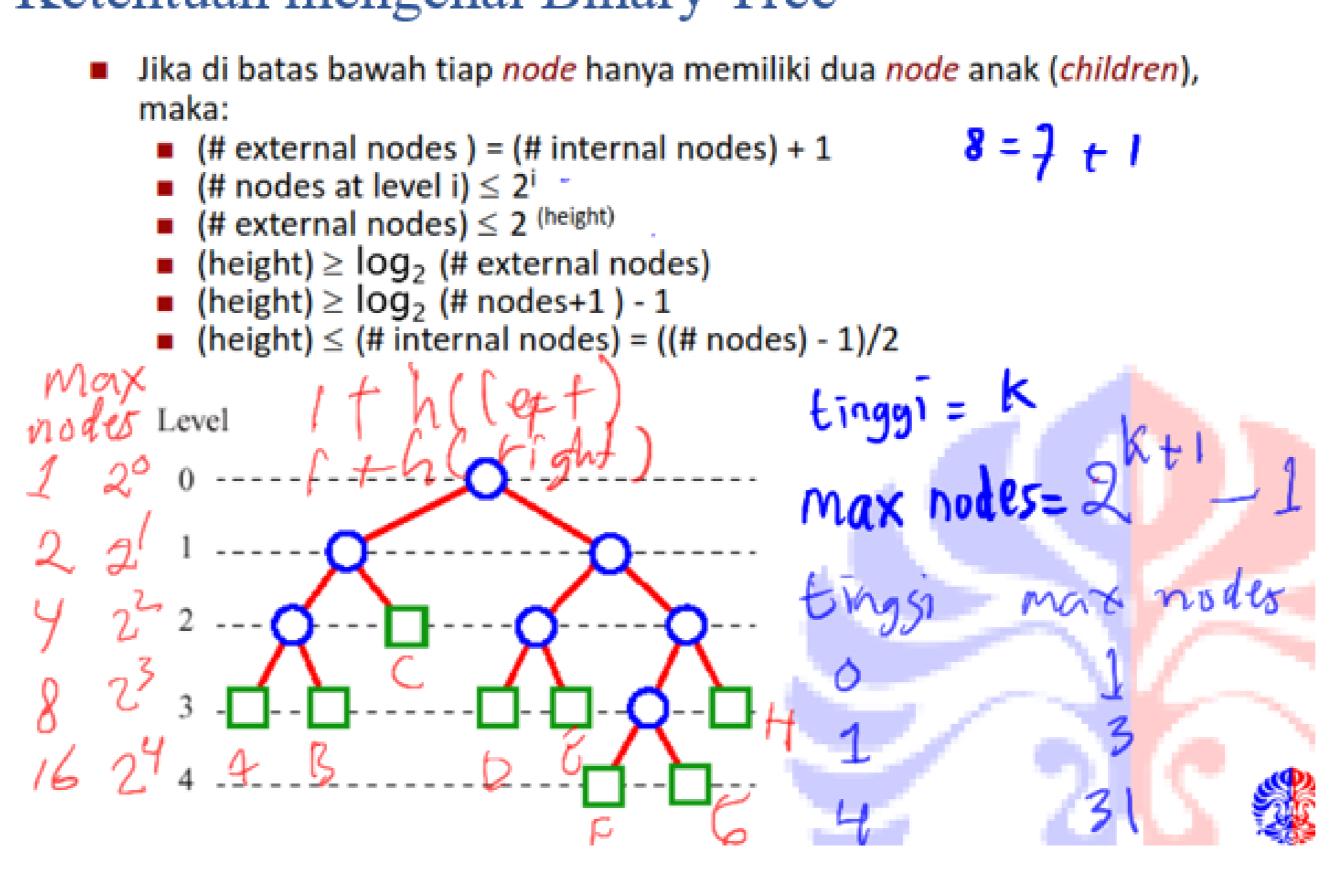
Allen Weiss, Florida International University. — Fourth edition.



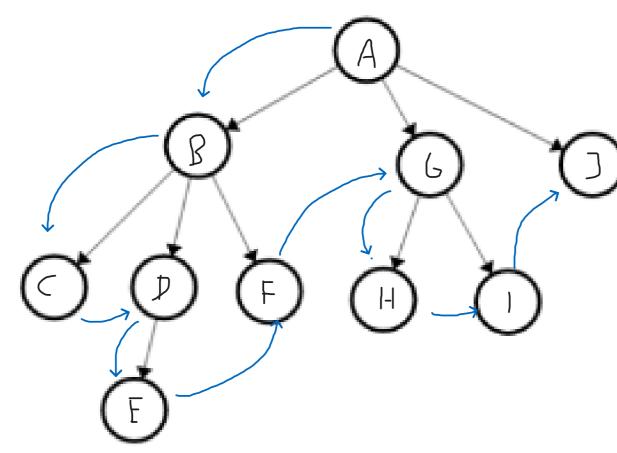
23. Tinggi sebuah binary tree adalah 0 jika hanya memiliki satu node yaitu root saja. Diketahui suatu binary tree tidak berisi internal node berdegree 1. Jika banyaknya node adalah 17 maka tinggi minimal binary tree tersebut adalah .....

```
Height >= log_2(17+1)-1
height >= 3.16
atau
\max nodes = 2^{(K+1)}-1
jika k = 3, max nodes = 15 nodes
Jika k = 4, max nodes = 33 nodes
```

## Ketentuan mengenai Binary Tree

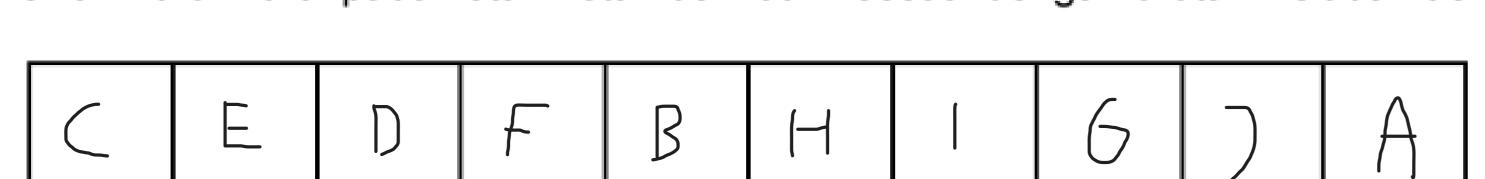


24. suatu general tree dengan struktur sebagai terlihat pada gambar di bawah ini.

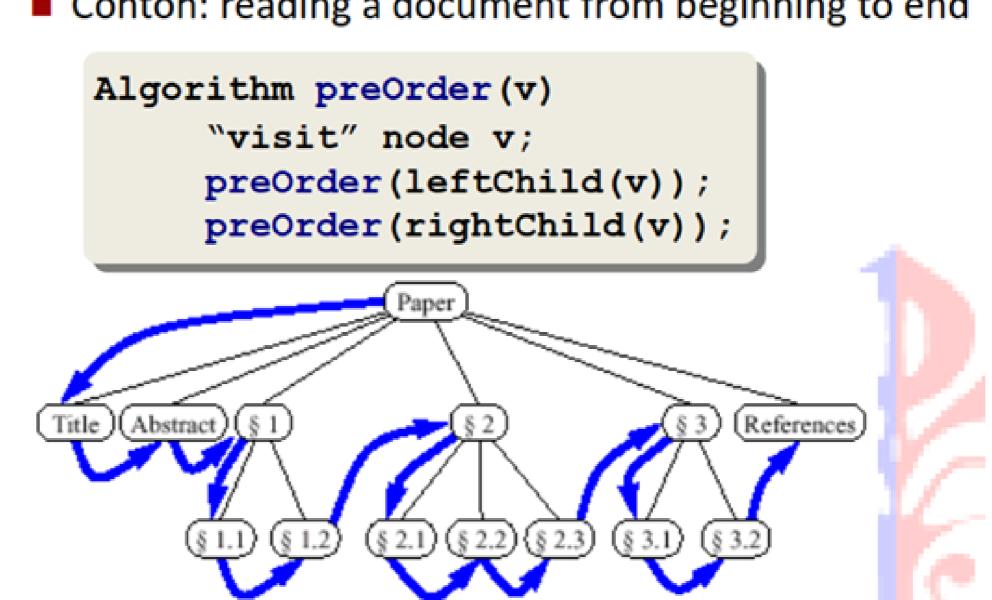


Setiap nodenya berisikan satu huruf unik yang saat ini tidak diketahui. Yang diketahui, hasil pre-order traversal-nya berupa string dari urutan huruf-huruf node sesuai urutan visit dalam traversal. Urutan children dalam traversal dari kiri ke kanan sesuai gambar. String keluaran tersebut adalah ABCDEFGHIJ. Jika algoritma diganti dengan post-order traversal, apa string keluarannya (dengan urutan children yang sama dan dicetak sebagai string huruf-huruf tanpa separator)?

Isikan huruf-huruf pada kotak-kotak berikut ini sesuai dengan urutan visit dari traversal.

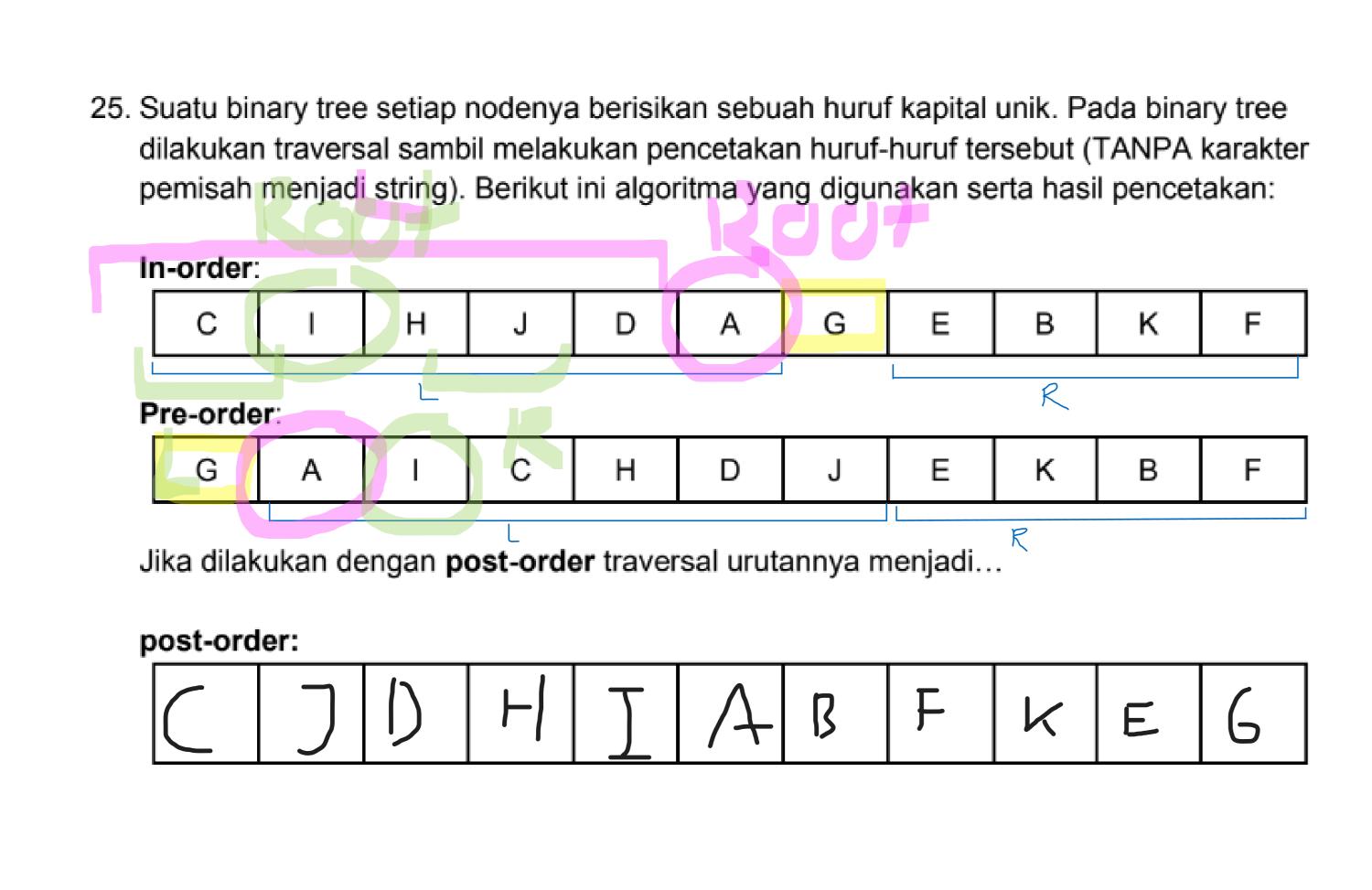


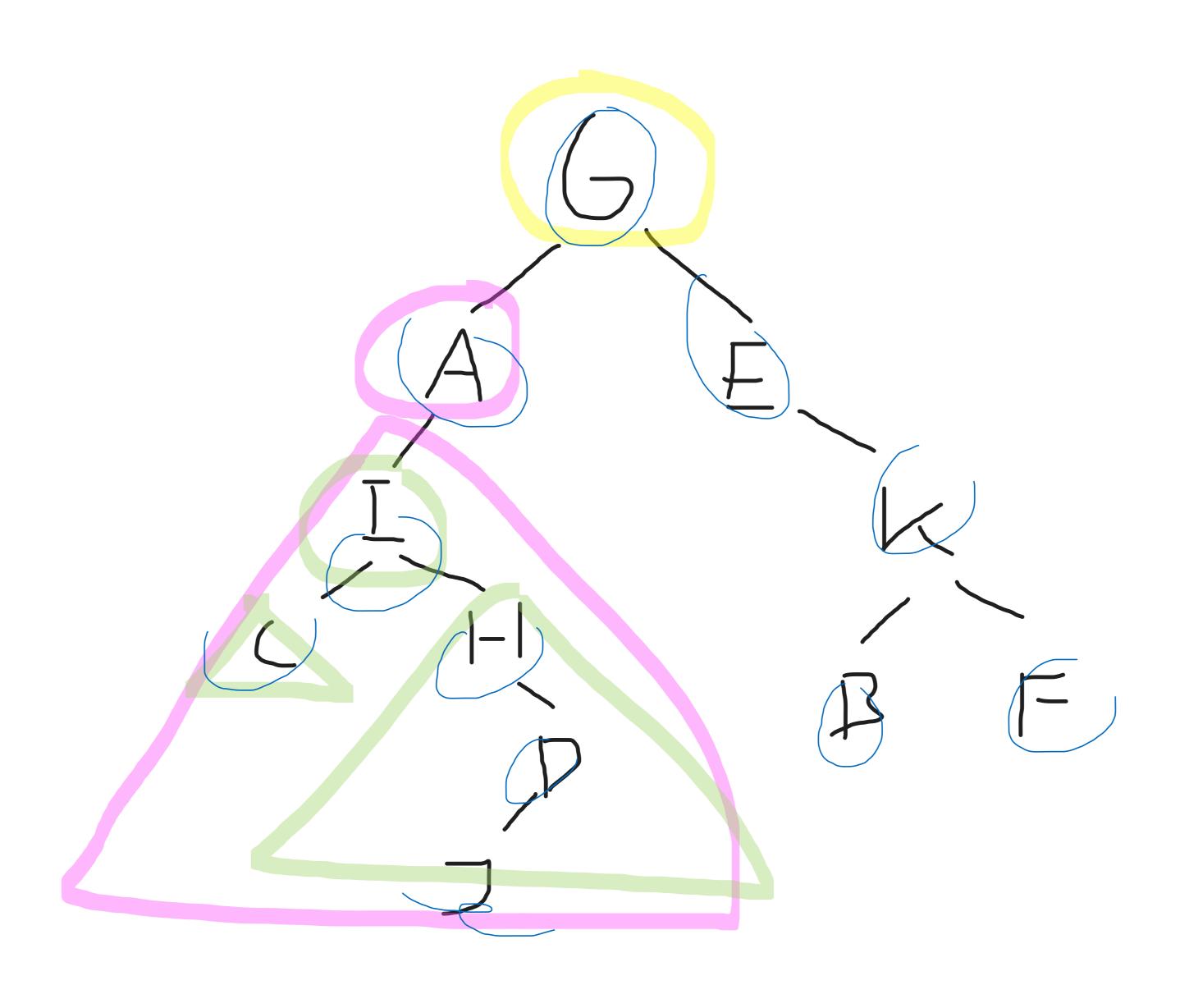
Contoh: reading a document from beginning to end



- Traversing Tree lainnya seperti level order
  - Preorder traversal -> mengunjungi root kemudian pohon kiri dan pohon kanan

  - o Inorder  $\rightarrow$  L Root R  $\circ$  Postorder  $\rightarrow$  L R Root





4. Binary tree didefinisikan dengan class node MyBTNode. Di dalam class ini ada method sumOfValueInDegree2Node() untuk menjumlahkan setiap bilangan di dalam field value setiap node berdegree-2. Method tsb belum lengkap dan anda harus melengkapinya sebagai modifikasi dari salah satu algoritma traversal (boleh mana saja). [hints: node ber-degree-2 adalah jika kedua cabangnya tidak null; code harusnya berisi sekitar 12 baris perintah].

```
class MyBTNode{
    int value;
    MyBTNode left, right;
    int sumOfValueInDegree2Node(){
        int sum = 0;
        if (left != null && right != null){
            sum += value;
        if (left != null){
            sum += left.sumOfValueInDegree2Node();
        if (right != null){
            sum += right.sumOfValueInDegree2Node();
        return sum;
```