



Binary Search Tree (BST)

- Binary Search Tree (pohon telusur biner)
- Disebut juga *Ordered Binary tree* yaitu binary tree yang seluruh **children** dari tiap **node** terurut.
- Data pada subtree kiri lebih kecil dari data pada subtree kanan.



- t = 20
-
- ```
graph TD; 20((20)) --- 10((10)); 20 --- 40((40)); 10 --- 6((6)); 10 --- 15((15)); 6 --- 2((2)); 6 --- 8((8)); 40 --- 30((30)); 30 --- 25((25));
```

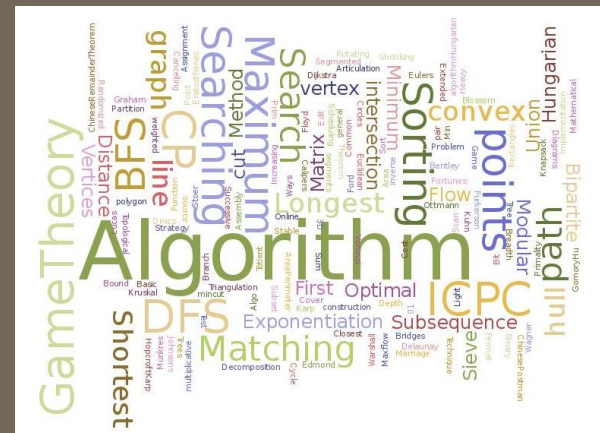
- 



# Latihan

- Buatlah sebuah Binary Search Tree berdasarkan proses-proses berikut :  
insert 17, insert 8, insert 32, insert 6, insert 10, insert 21, insert 45, insert 1, insert 7, insert 9, insert 26, insert 37, insert 50.





# Binary Tree Traversal

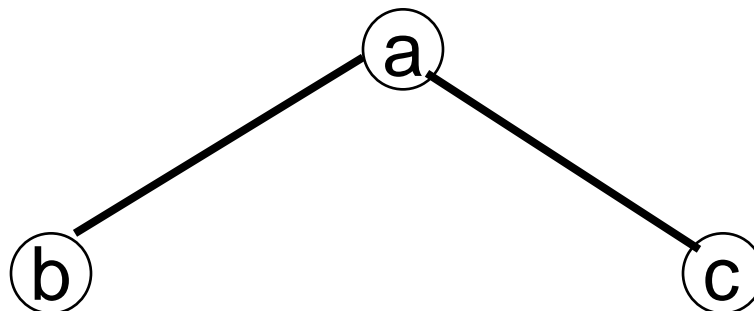


# Definisi

- Penelusuran seluruh node pada binary tree.
- Metode :
  - Preorder
  - Inorder
  - Postorder
  - Level order





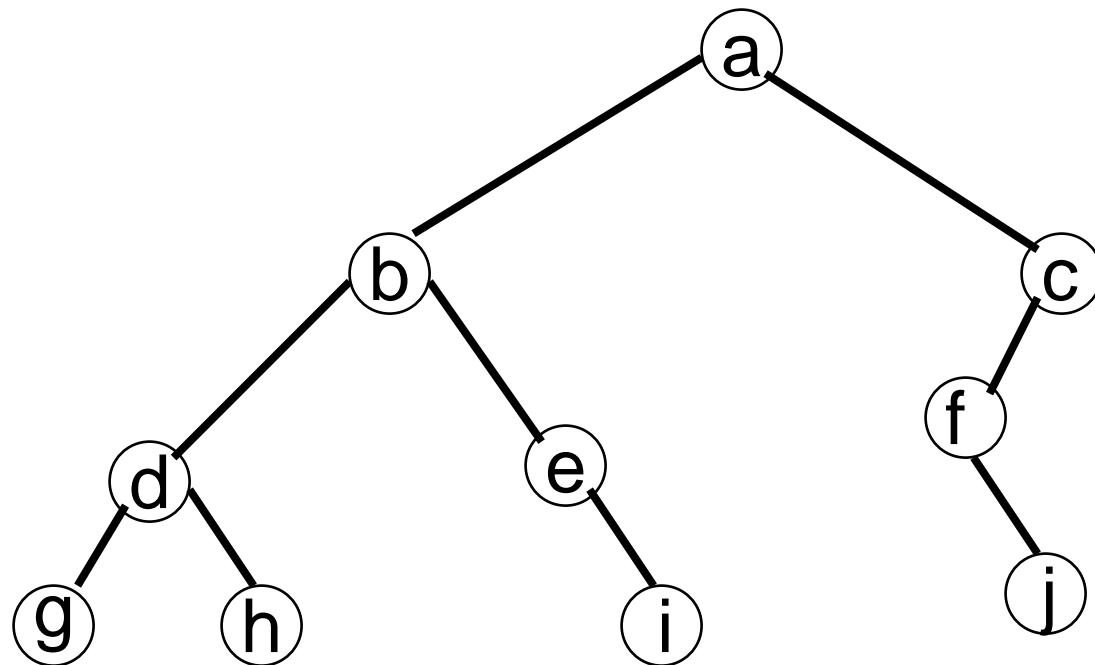


a b c





## Preorder Example (visit = print)

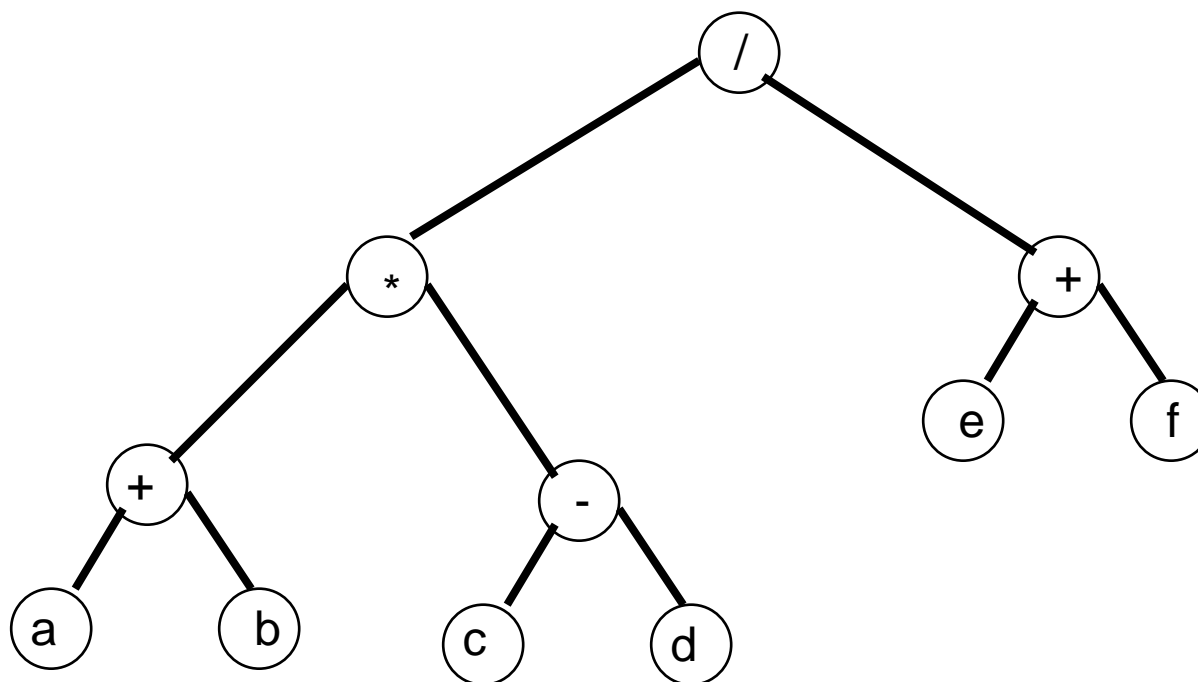


a b d g h e i c f j





# Preorder Of Expression Tree



/ \* + a b - c d + e f

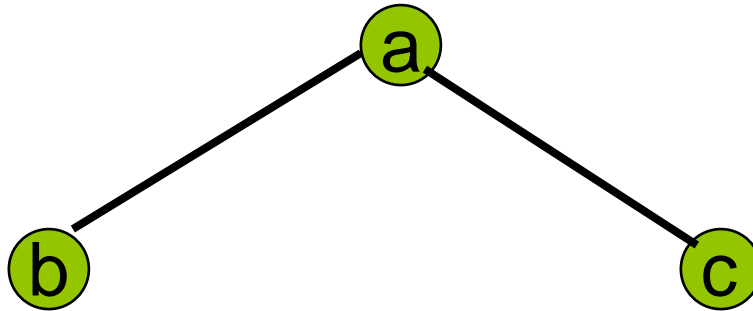
Gives prefix form of expression!







# Inorder Example (visit = print)

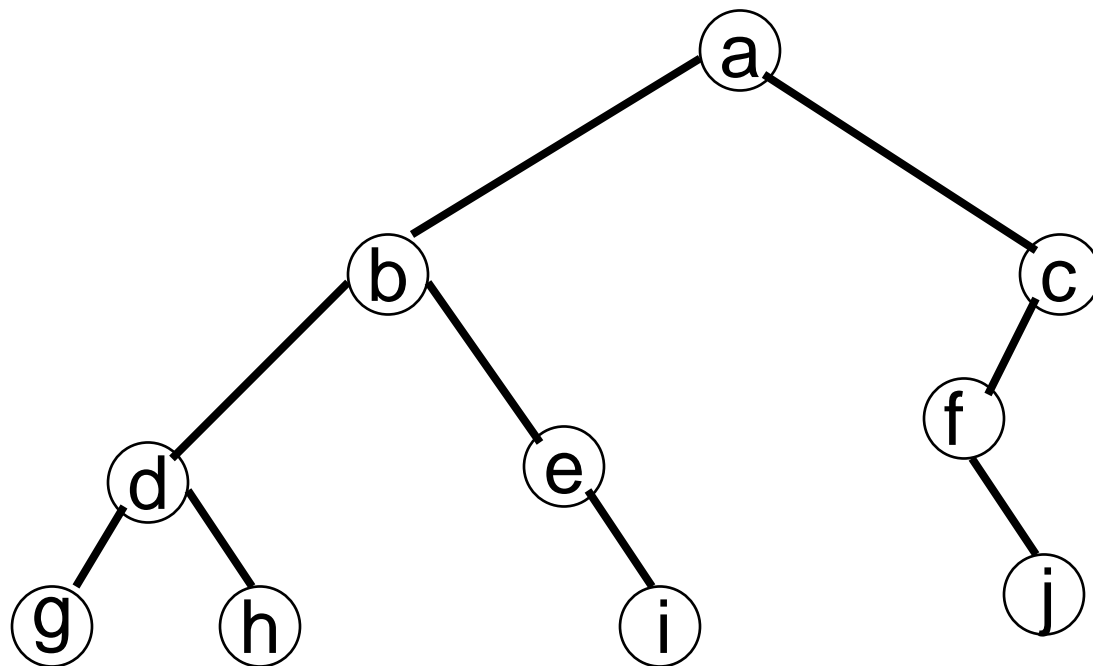


b a c





## Inorder Example (visit = print)



g d h b e i a f j c

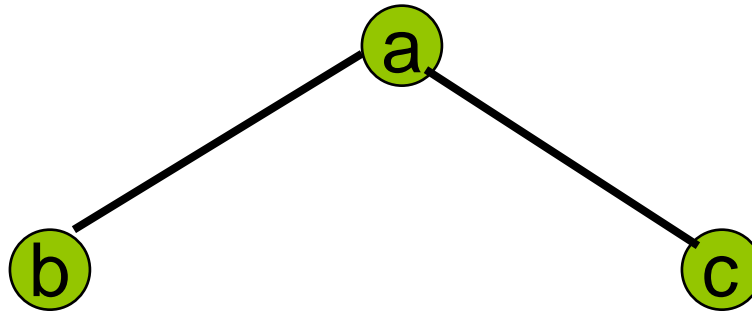




1. Secara rekursif mencetak seluruh data pada subpohon kiri
2. Secara rekursif mencetak seluruh data pada subpohon kanan
3. Cetak data pada root



## Postorder Example (visit = print)

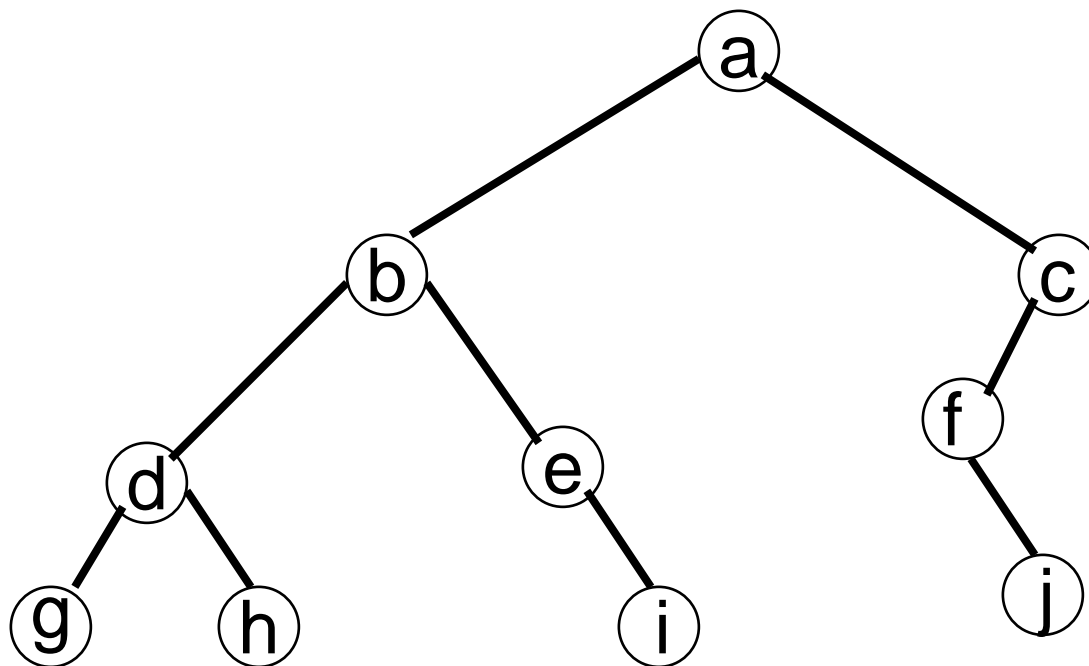


b c a





## Postorder Example (visit = print)



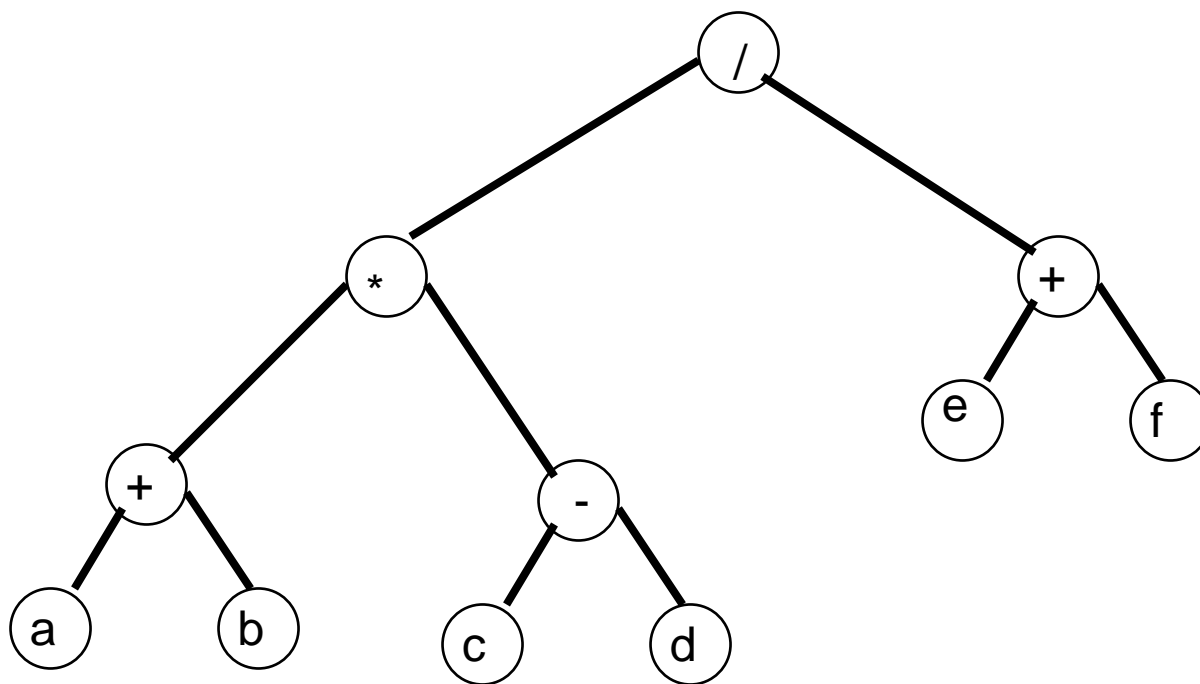
g h d i e b j f c a







# Postorder Of Expression Tree



a b + c d - \* e f + /





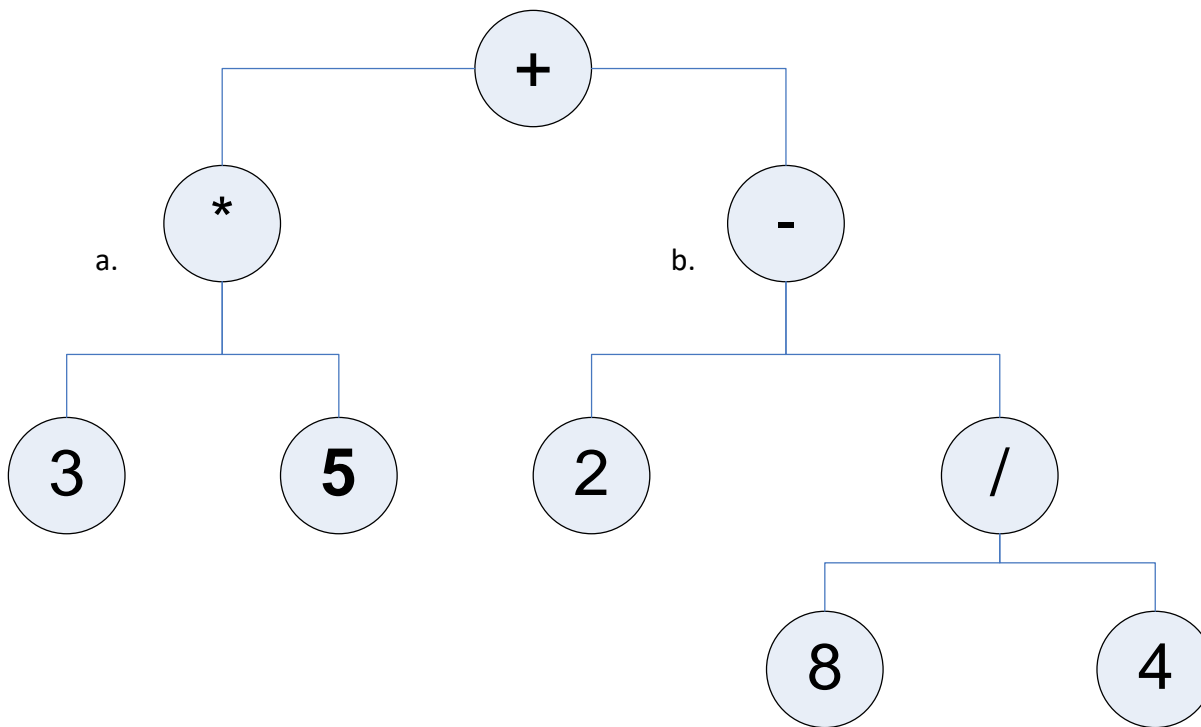
# Latihan

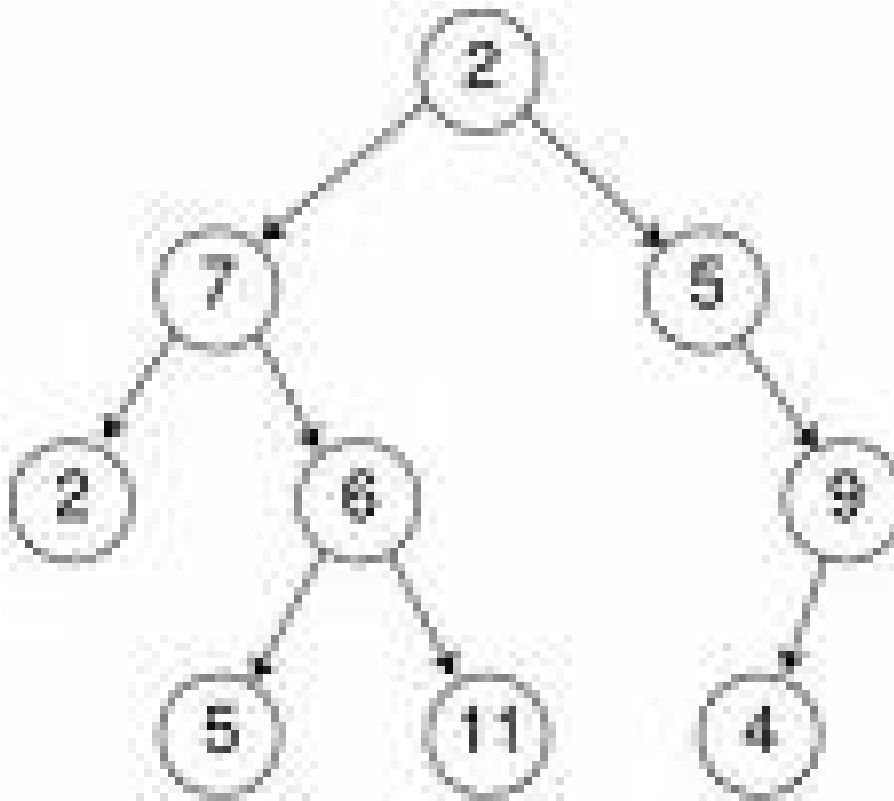
- Telusuri pohon biner berikut dengan menggunakan metode pre, in, post, dan level traversal.



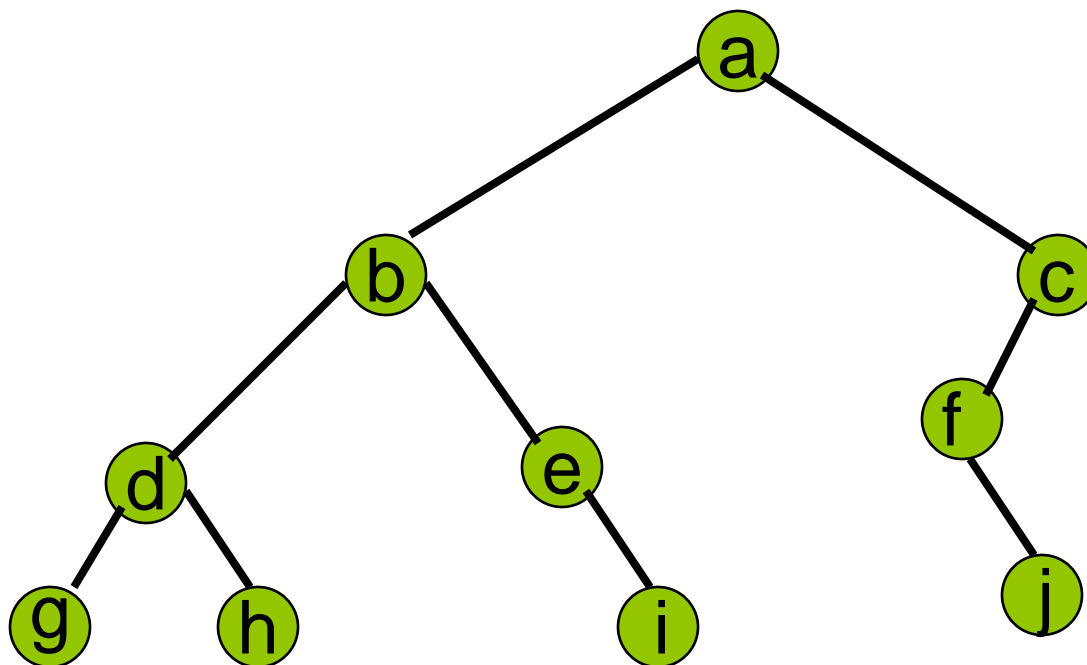


# Latihan 1



[illegible]

## Level-Order Example (visit = print)



a b c d e f g h i j

