Binary Search Tree

Pohon Biner Terurut

Binary Search Tree is a node-based binary tree data structure which has the following properties:

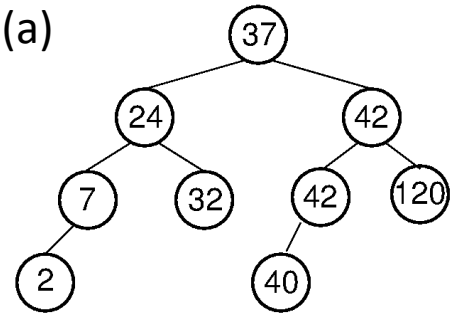
- The left subtree of a node contains only nodes with keys lesser than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.

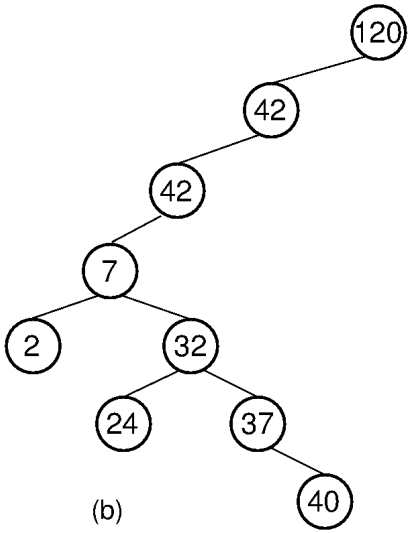
INSTITUT
TEKNOLOGI
HARAPAN
BANGSA

Veritas vos liberabit

Tunjukkan mana yang BST dan mana yang bukan, beserta alasannya

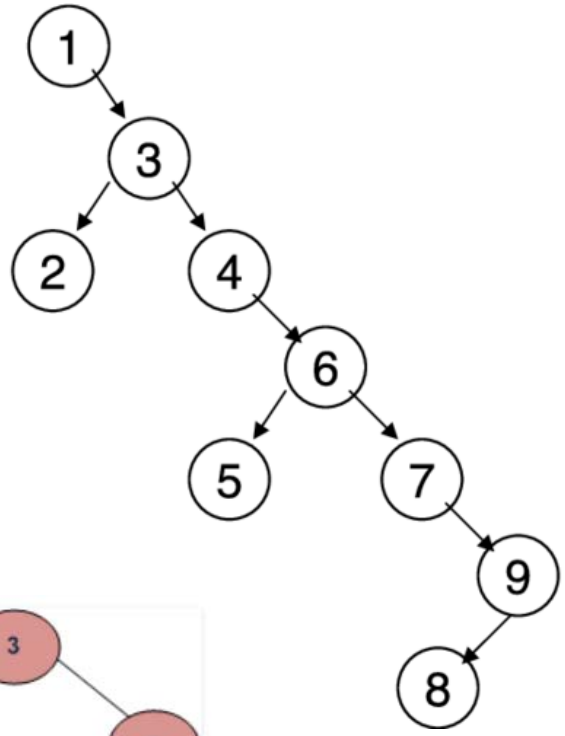
(a)



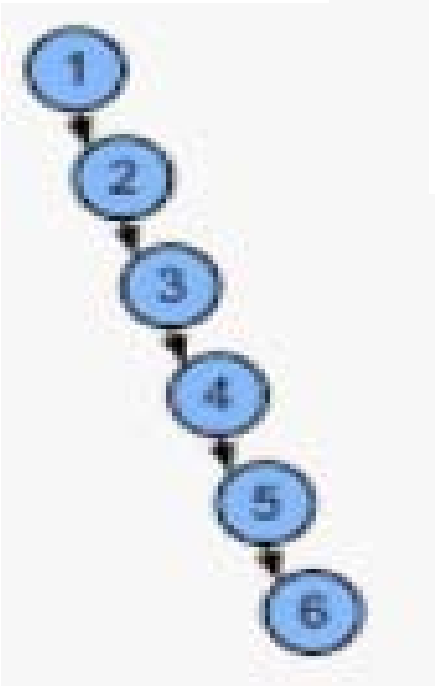


(b)

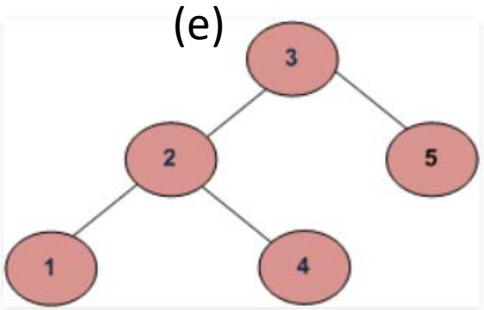
©



(d)



(e)

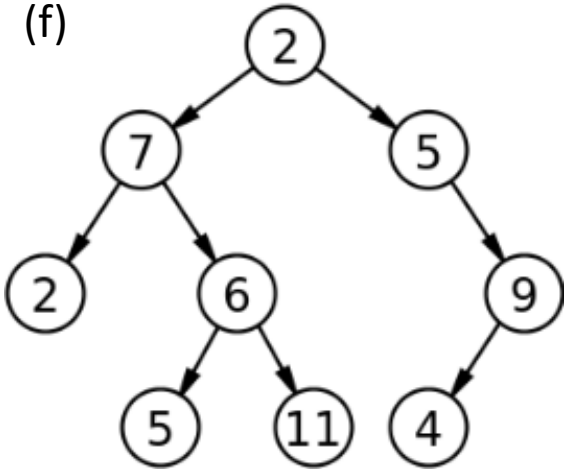


INSTITUT TEKNOLOGI HARAPAN BANGSA

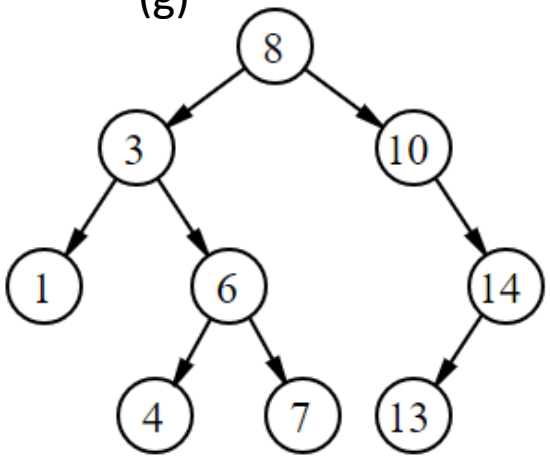
Veritas vos liberabit

Tunjukkan mana yang BST dan mana yang bukan, beserta alasannya

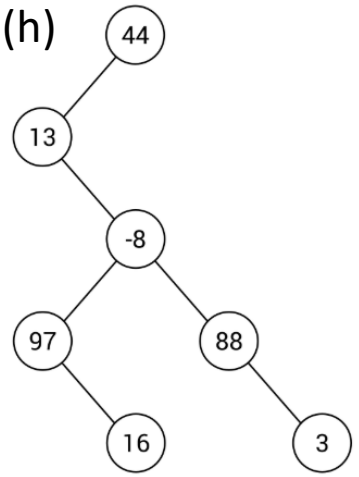
(f)



(g)



(h)



Algoritma INSERT Rekursif

```
Insert(X, T)
If T = nil
Then {insert}
        new(T); Val(T) = x; left(T) = nil, right(T) = nil
Else
        case
                x < val(T) : Insert(X, left(T))
                x > val(T) : Insert(X, right(T))
                x = val(T) : {duplikasi}
        endcase
endif
```

INSTITUT TEKNOLOGI HARAPAN BANGSA

Veritas vos liberabit

Algoritma INSERT NON-Rekursif

```
Insert(X, T)
If T = nil
Then {insert}
        new(T); Val(T) = x; left(T) = nil, right(T) = nil
Else
        P = T
        repeat
                father = P
                if  x<val(P) then P = left(P) else P = right(P)
        until P = nil
        new(P); Val(P) = x; left(P) = nil, right(P) = nil
        if  x < val(father)
        then  left(father) = P
        else  right(father) = P
        endif
endif
```

INSTITUT TEKNOLOGI HARAPAN BANGSA

Veritas vos liberabit

Latihan INSERT

Diberikan data-data berikut, gambarkan BST yang terbentuk menggunakan algoritme REKURSIF dan NON-REKURSIF :

1. 7,4,8,3,2,1,9
2. 4,8,1,5,9,2,6

Masalah :

1. Apakah BST yang terbentuk sama jika urutan data berbeda?
2. Coba ubah urutannya dan tuliskan urutan yang Anda buat serta gambarkan BST nya

Algoritma SEARCH Rekursif

Search(T,X,found,P)

```
If T = nil
Then found = false
Else
        case
                x < val(T) : Search(left(T), X, found, P)
                x > val(T) : Search(right(T), X, found, P)
                x = val(T) : found = true; P = T
        endcase
endif
```

INSTITUT TEKNOLOGI HARAPAN BANGSA

Veritas vos liberabit

Algoritma SEARCH NON-Rekursif

Search(T,X,found,P)

```
Q = T ; found = false
While (Q <> nil) and (not found)
Do    case
            x < val(T) : Q = left(T)
            x > val(T) : Q = right(T)
            x = val(T) : found = true; P = Q
      endcase
endwhile
```

INSTITUT TEKNOLOGI HARAPAN BANGSA

Veritas vos liberabit

Efisiensi

- Similar to binary search in an array, the search speed is proportional to $\log_2(n)$, if the tree contains N node.
- However, if the tree is very skewed (left or right) the efficiency is similar to the efficiency of a sequential search

Data Non Unik

- Bagaimana jika data tidak unik, misal data 5 muncul 2 kali ?