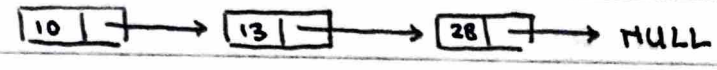
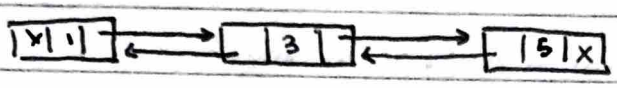
No. Vika Valencia . S Date

Linked List

1. Single



Double



Circular

2. Array

- Pakai indeks
- Sifatnya statis (ukurannya)
- Tau alamatnya dimana

Linked List

- Tidak pakai indeks
- Sifatnya fleksibel
- Tidak tau alamatnya dimana

3. Traverse until two pointers meet at the same node (Find loop)

```
Struct Node
    int data
    Node *next

Void make (struct Node **neww int data)
    Node * newnode = (Node**) malloc (sizeof (*Node))
    newnode → data = data
    newnode → next = * neww
    *neww : newnode

Int loop (Node *node)
    Node * slow = node , *fast = node
    while (slow AND fast AND fast→next)
        slow = slow→next
        fast = fast→next→next
        If (slow = fast) return 1
    return 0
```

Floyd's

KIKY

Date. :

Stack and Queue

1. Queue mempunyai size yang terbatas. Queue : FIFO (First in First out), Stack = FILO (First in last out)

2. Prefix → operator di depan
   Infix → Operasi hitung biasa → operator di tengah
   PostFix → operator di belakang
   → 1 stack isi semua operasi, baca sampai operator (paling penting), ambil 2 operand setelahnya, masukan ke stack lain, dihitung, yang distack satunya di pop lalu di taroh hasil hitungnya. Gitu sampai operatornya habis
   → Sama, cuman baca dari operator yang paling penting dan yang diambil 2 operand yang mengapitnya
   → sama, bedanya baca dari belakang dan yang diambil 2 operand dari depan

Pre	In	Post
+ 5 * 3 4	5 + 3 * 4	5 3 4 * +

Hashing and Hash tables

1. Hash table is where we store the original string
   Hash function : operasi yang dijalankan
   Collision : eror / ada 2 atau lebih keys yang menghasilkan hash yang sama

2. Linear Probing : search the next empty slot
   Chaining : put the next string as a chained list

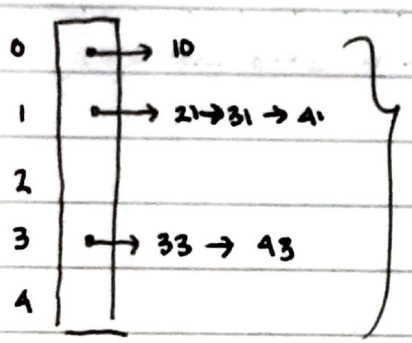
0	4
1	9
2	10
3	

insert 10 % 4 : 2
insert 9 % 4 : 1
insert 4 % 4 : 0

} Linear probing

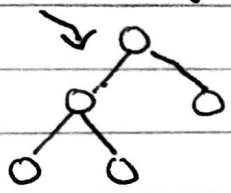
No. :

Date. :

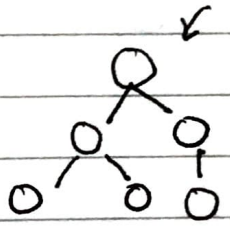


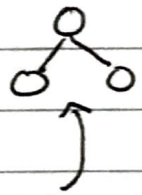
Binary Search Tree

1. Full binary tree : punya 0 / 2 anak



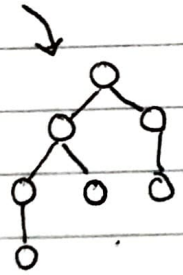
2. Complete binary tree : semuanya wajib terisi node kecuali height paling bawah



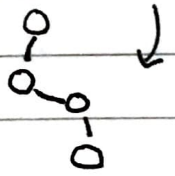


- Perfect binary tree : node internal harus ada 2 anak, dan harus sama levelnya

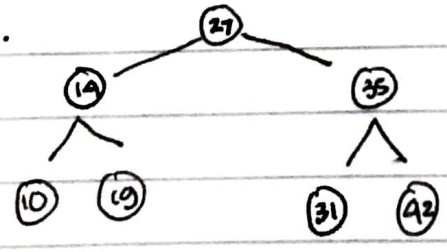
- Balanced binary tree : tinggginya O(log N) . N = jumlah node. Subtree kiri dan kanan edgenya selisih 1



- Degenerate : setiap internal node anaknya 1

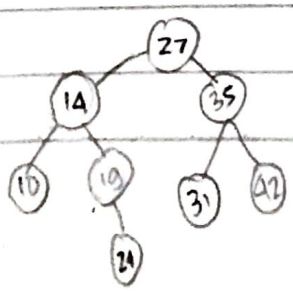


2.

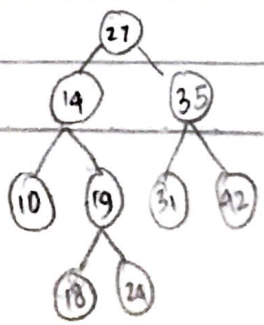


Insertion 24 : cek jika dia < root maka kekiri, jika > root ke kanan

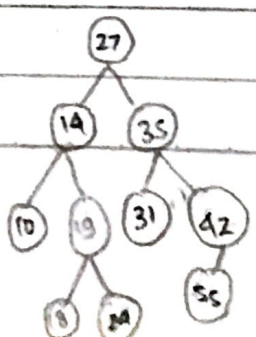
- 24



- 18



- 55

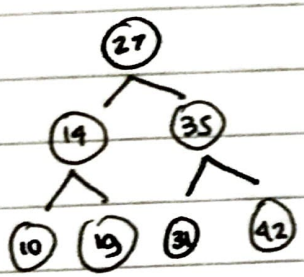


KIKY

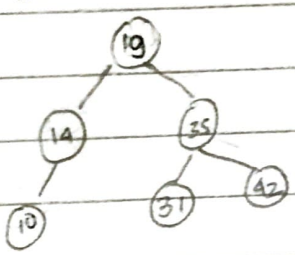
cari ke kiri jika yang dihapus < root
kanan > root

No. : Date. :

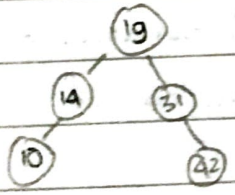
3. Deletion : jika di hapus maka posisinya diganti yang lain yang lebih cocok



27



35



42