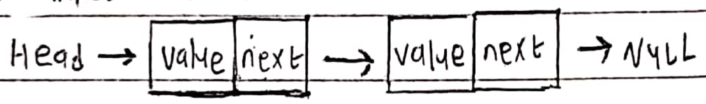
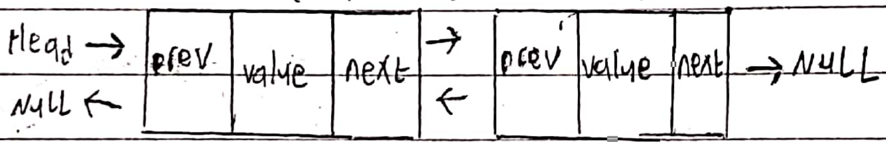
Name: c_c_c_c

Linked list

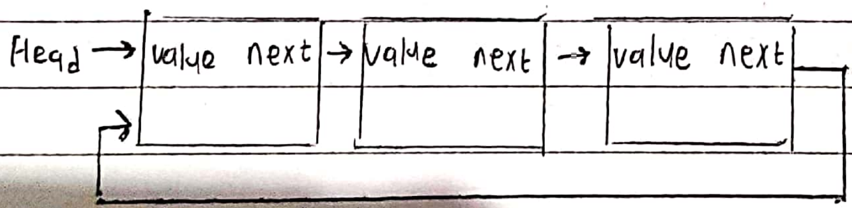
1) * Single linked list (1 pointer)



* Double linked list (2 pointer)



* Circular linked list (last element linked to first element)



2) Linked list	Array
Slow search time	fast search time
Dynamic size	Static size
More memory (pointer needed)	Less memory
fast insertion/deletion	Slow insertion/deletion
effecient memory allocation	Inefficient memory allocation

) the floyd algorithm is to find shortest distance between every pair of vertices in a given edge weighted directed graph

```
Start
    for i ← 1 to n do
        for j ← 1 to n do
            for k ← 1 to n do
                if Mfloyd (i,j) ≥
                    (mfloyd [i,k] + M floyd (k,j) then
                        Mfloyd [i,j] =
                            mfloyd (i,k) + M floyd (k,j)
```

Stack and Queue

1)

Stack	Queue
Last in first out	First in first out
using push and pop operation	using enqueue and dequeue
using one pointer	using more than one pointer
Same end is to insert	One end is for insertion and
and also delete value	the other end is for deletion

2) infix = operator is written between operand $(4 * 10)$
prefix : operator ditulis sebelum operand $(* 4 10 + 5 * 3 4)$
postfix: operator is writen after operand $(4 10 *, 5 3 4 * +)$

+ prefix	+ post fix (search for 2 variable behind operator)
$2+3*(5-2)/3$	$8\ 5\ 3 * 2\ 3 + - +$ scan until reach the first operator
$2+3*3/3$	$8\ 15\ 2\ 3 + - +$ (5.3)
$2+9/3$	$8\ 15\ 5 - +$ $(2+3)$
$2+3$	$8\ 10 +$ $(15-5)$
6	18 $(8+10)$
Iterate the	
highest predence	
and total it	

Hashing and Hash tables

1) Hash function means → kita mengekripsi sesuatu text sesuai function yang kita bikin
ada sebuah text → masukkan ke hash function → hashed text

hash table → Dimana kita store original stringnya (kita harus tentukan dulu besar hashtablenya)

Collision → bentrokan dimana ada dua nama yang sama misal huruf depannya
→ contoh Darnell dan data berikutnya Doni
ini bisa diatasi dengan linear probing and chaining

2) - Linear probing
- Linear chaining (linked list)

Linear probing (search for the empty slot).

Darnel $\rightarrow$

0	Andi
1	Budi
2	Dana
3	Dodi
4	

ternyata 2 penuh maka loop for the next empty slot

Darnel $\rightarrow$

0	Andi
1	Budi
2	Dana
3	Dodi
4	Darnel

di index 4 kosong

Linear chaining (put the string as chain linked list)

Caca $\rightarrow$

0	andi		
1	Bana	Budi	
2	candi		
3	Dadi	Dado	

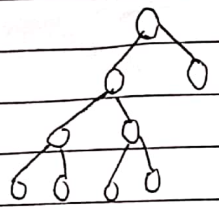
masuk ke index 2 karena sama depannya dan taro di chainnya

caca $\rightarrow$

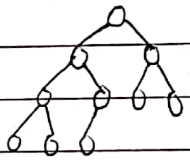
0	andi		
1	Bana	Budi	
2	Candi	caca	
3	Dadi	Dado	

Binary Search Tree

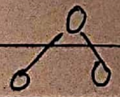
1) - Full binary tree (punya 0 / 2 anak, tak mungkin 1)



- Complete binary tree (harus keisi semua node kecuali yang paling bawah)

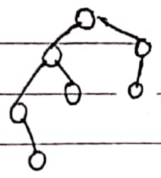


- Perfect binary tree (Setiap Internal node hanya boleh punya 2 anak dan leafnya di level yang sama)

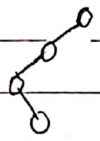


Scanned with CamScanner

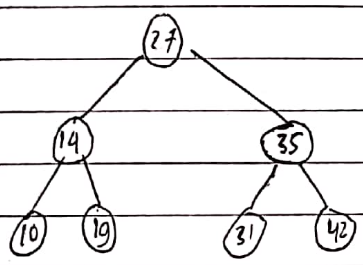
- balanced binary tree (tinggi O log n, dimana n itu nodes, sub tree kiri dan sub tree kanan selisihnya satu)

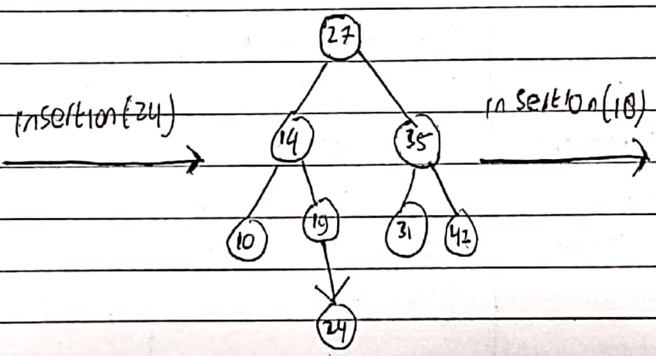


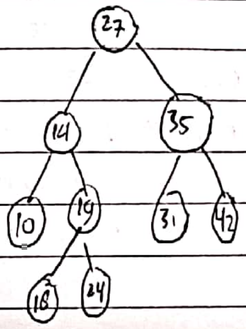
- Degenarate Binary tree (Internal node hanya cuman single node seperti linked List (satu arah)

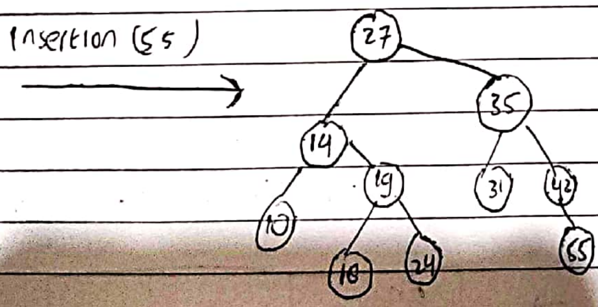


2)







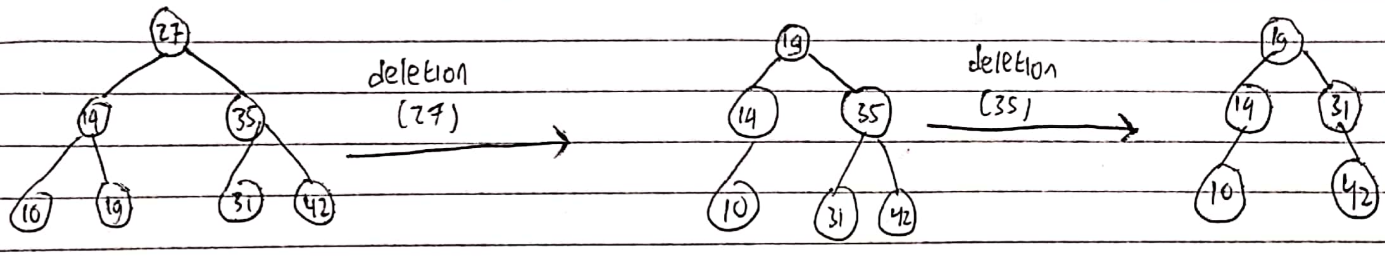


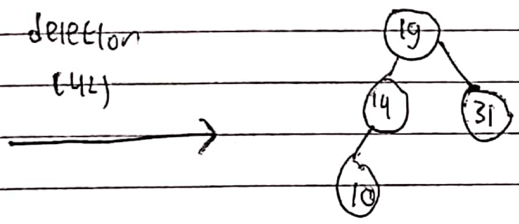
* insertion (24) → Subtree kiri karena lebih kecil dari 27 terus ke sub tree kanan karena lebih besar dari 19 dan langsung connect ke bawah 19 karena 19 single

* insertion (18) → ke subtree kiri karena lebih kecil dari 27 terus ke sub tree kanan karena lebih besar dari 19 dan ke sub tree kiri karena lebih kecil dari 19

* insertion (55) → ke sub tree kanan karena lebih besar dari 27 dan ke sub tree kanan karena lebih besar dari 35 dan langsung connect ke bawah 42 karena single

3)





* deletion (27) → pilih antara kiri terbesar (subtree) dan kanan terkecil
kasus ini kiri terbesar yaitu 19, maka 27 ditimpa dengan 19

* deletion (35) → karena di subtree kanan maka pilih calon yang terkecil = 31
35 ditimpa dengan 31

* deletion(42) → delete biasa saja ditranverse (search) dan pop