KUNCI JAWABAN
QUIZ BEFORE UTS
COMP6362004 – DATA STRUCTURES (LB20)
EVEN 2023/2024

1. **[LO 2; Weight 20%]** Given an infix expression:

A * (B / C) – D ^ E / (F * G)

Convert the following infix to postfix and to prefix. Show the results and the process. Use one of the methods you have understood and learned in class.

Answer:

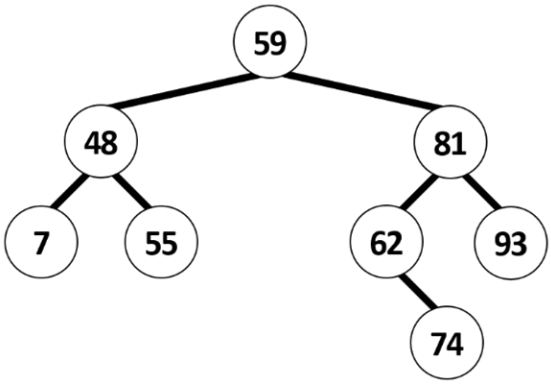
Manual:

Infix to Prefix	Infix to Postfix
A * (B / C) – D ^ E / (F * G)	**A * (B / C) – D ^ E / (F * G)**
➢ **A * / B C – D ^ E / * F G**	➢ **A * B C / – D ^ E / F G ***
➢ **A * / B C – ^ D E / * F G**	➢ **A * B C / – D E ^ / F G ***
➢ *** A / B C – ^ D E / * F G**	➢ **A B C / * – D E ^ / F G ***
➢ *** A / B C – / ^ D E * F G**	➢ **A B C / * – D E ^ F G * /**
➢ **– * A / B C / ^ D E * F G**	➢ **A B C / * D E ^ F G * / –**

Stack: **Infix to Postfix**

Input String	Output Stack	Operator Stack
A*(B/C)-D^E/(F*G)	A	
A*(B/C)-D^E/(F*G)	A	*
A*(B/C)-D^E/(F*G)	A	*(
A*(B/C)-D^E/(F*G)	AB	*(
A*(B/C)-D^E/(F*G)	AB	*(/
A*(B/C)-D^E/(F*G)	ABC	*(/
A*(B/C)-D^E/(F*G)	ABC/	*
A*(B/C)-D^E/(F*G)	ABC/*	-
A*(B/C)-D^E/(F*G)	ABC/*D	-
A*(B/C)-D^E/(F*G)	ABC/*D	-^
A*(B/C)-D^E/(F*G)	ABC/*DE	-^
A*(B/C)-D^E/(F*G)	ABC/*DE^	-/
A*(B/C)-D^E/(F*G)	ABC/*DE^	-/(
A*(B/C)-D^E/(F*G)	ABC/*DE^F	-/(
A*(B/C)-D^E/(F*G)	ABC/*DE^F	-/(*
A*(B/C)-D^E/(F*G)	ABC/*DE^FG	-/(*
A*(B/C)-D^E/(F*G)	ABC/*DE^FG*	-/
A*(B/C)-D^E/(F*G)	ABC/*DE^FG*/-	

2. **[LO 2; Weight 25%]** Given Tree as follows:



a. Write down the root node, leaf node and level of the tree above!

Answer:

- root node: 59
- leaf node: 7, 55, 74, 93
- level of the tree: 3

b. If implemented with an array, draw the complete array of array elements for the Tree along with the index position of each element.

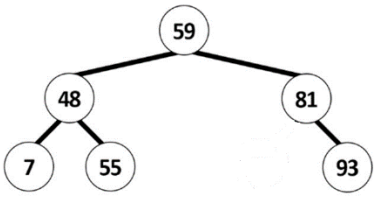
Answer:

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
node	59	48	81	7	55	62	93						74		

c. It is assumed that the tree above is a **Binary Tree**.

Describe the last condition after **deleting 62**.

Answer:



Penjelasan meliputi level dan degree yang dimiliki tree setelah deletion 62, tipe binary tree yang terbentuk.

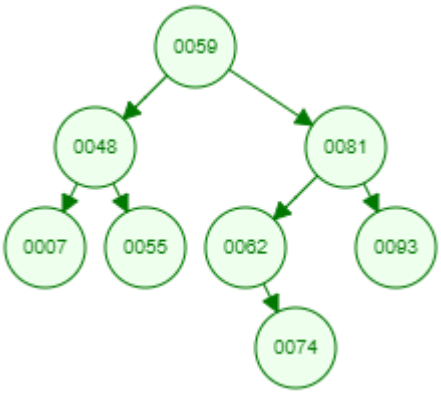
d. It is assumed that the tree above is a **Binary Search Tree**.

Simulate Binary Search Tree shape changes in each of the following operations:

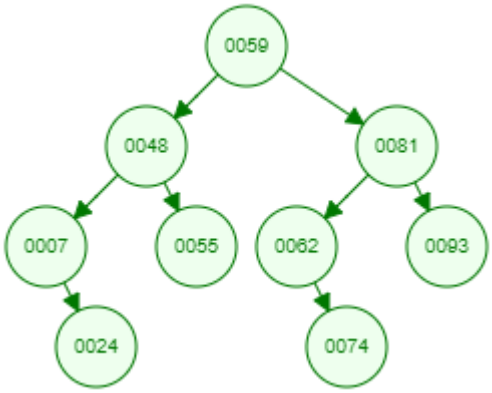
insert 24, **insert 11**, **insert 29**, **insert 36**, **delete 48**, **delete 81**, and **delete 59**

Answer:

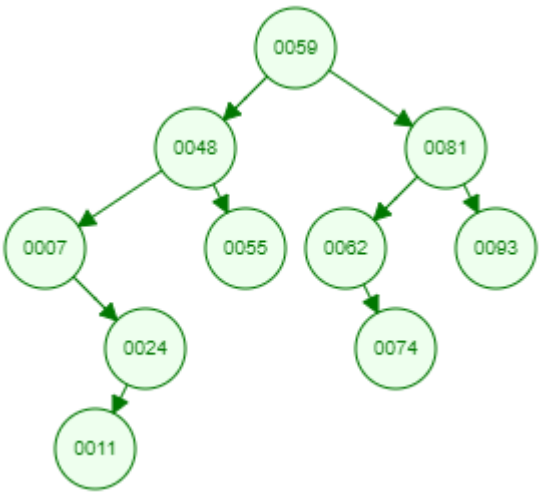
1. Kondisi awal:



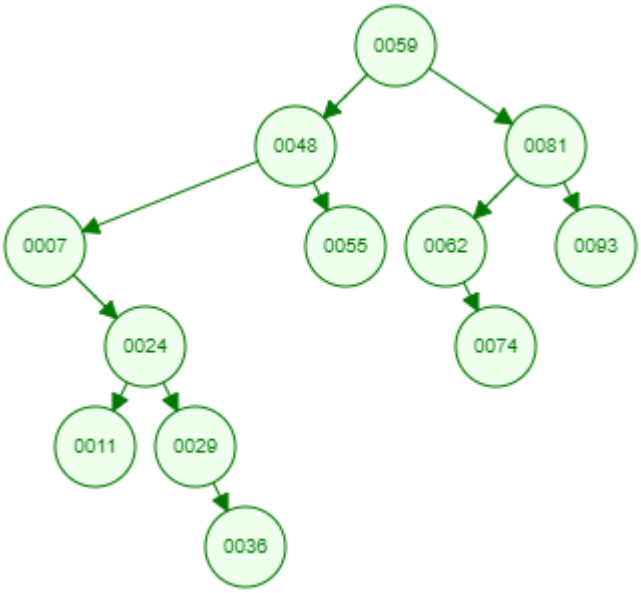
2. insert 24



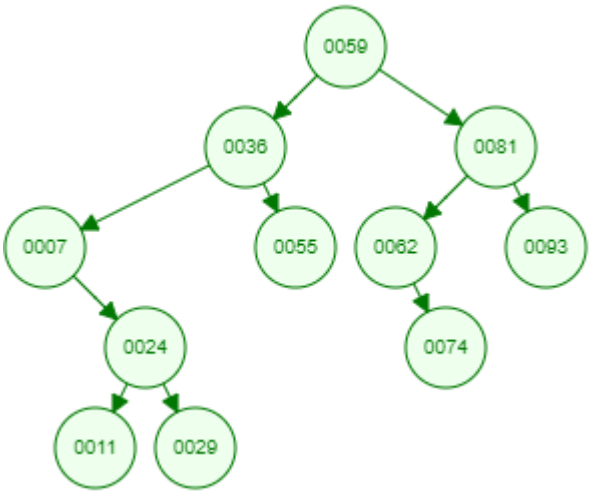
3. insert 11



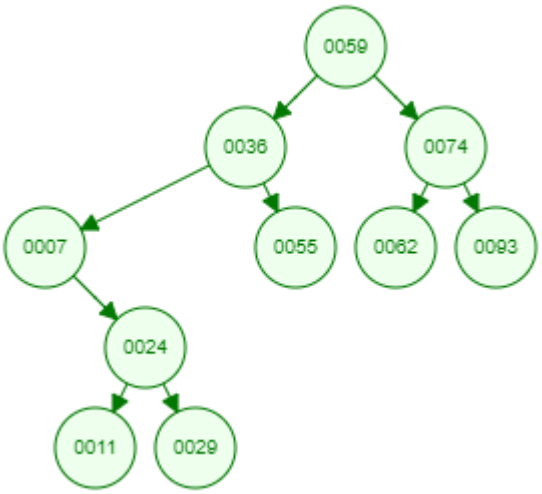
5. insert 36

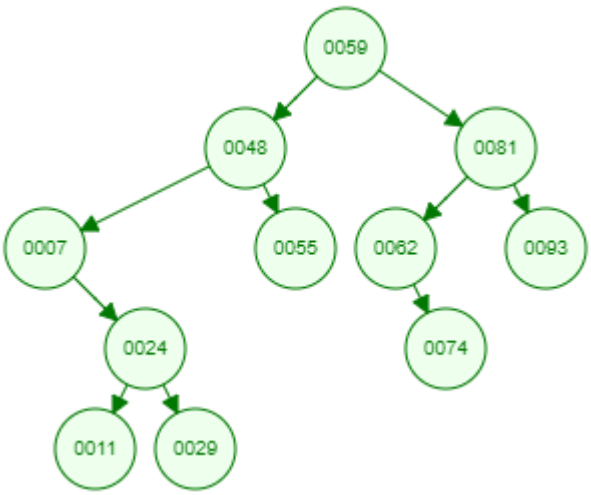
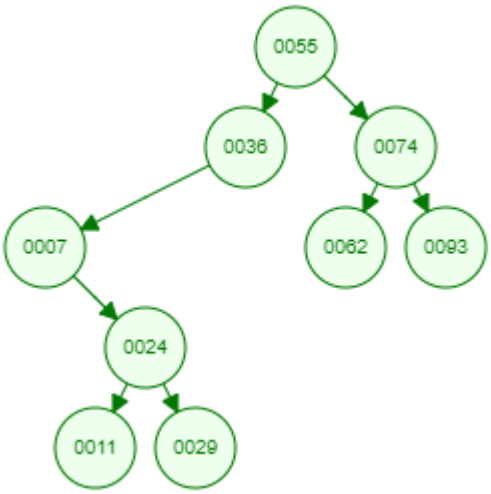


6. delete 48



7. delete 81



4. **insert 29**	8. **delete 59**
	

Note: Replacement nodes that are removed priority is taken from the left subtree first, if not available take the replacement from the right subtree

e. Write the traversal results using **PreOrder**, **InOrder**, **PostOrder** from the Binary Search Tree that has been created

Answer:

PreOrder **:** 55, 36, 7, 24, 11, 29, 74, 62, 93

InOrder **:** 7, 11, 24, 29, 36, 55, 62, 74, 93

PostOrder **:** 11, 29, 24, 7, 36, 62, 93, 74, 55

3. **[LO 2; Weight 10%]**

Explain the differences between Tree, Binary Search Tree, and Trie!

Answer:

Jawaban meliputi keyword berikut

- **Tree** : childen Tree dapat lebih dari 2 child; tidak ada aturan mengenai penempatan children; Tree secara umum dapat berupa balanced Tree, skewed Tree, atau jenis yang lain.
- **Binary Search Tree**: tree yang maksimum memiliki 2 anak, yaitu anak kiri & anak kanan; binary tree dimana node pada anak kiri < root sedangkan node pada anak kanan > root
- **Trie**: tree yang dapat memiliki > 2 anak. Pohon terurut untuk menyimpan suatu himpunan string dimana setiap node pada pohon tersebut mengandung awalan (*prefix*) yang sama, trie disebut juga pohon prefix.

4. **[LO 2; Weight 20%]** Consider there is a hash table with size = 100.
Using Mid Square, Folding, and Division method, show the hashing process for the following key: 150, 121, 2500

Answer:

- metode Mid Square:
  - untuk key 150

    maka, 150^2 = 22500, misalnya yang diambil adalah mulai digit ke-2 s/d digit ke-3 yaitu 25 sehingga hash value-nya adalah **25** atau digit ke-3 s/d digit ke-4 yaitu 50 sehingga hash value-nya adalah **50**
  - untuk key 121

    121^2 = 14641

    digit ke-2 s/d digit ke-3 = **46**, atau

    digit ke-3 s/d digit ke-4 = **64**
  - untuk key 2500

    2500^2 = 6250000

    digit ke-3 s/d digit ke-4 = **50**, atau

    digit ke-4 s/d digit ke-5 = **0**
- metode Folding (sum):
  - untuk key 150

    diumpamakan yang di-fold adalah setiap 2 digit lalu dijumlahkan (sum)

    150 = 15 dan 0 → 15 + 0 = 15 sehingga hash value-nya adalah 15
  - untuk key 121

    diumpamakan yang di-fold adalah setiap 2 digit lalu dijumlahkan (sum)

    121 = 12 dan 1 → 12 + 1 = 13 sehingga hash value-nya adalah 13
  - untuk key 2500

    diumpamakan yang di-fold adalah setiap 2 digit lalu dijumlahkan (sum)

    2500 = 25 dan 00 → 25 + 0 = 25 sehingga hash value-nya adalah 25
- metode Division (misal mod 100):
  - untuk key 150

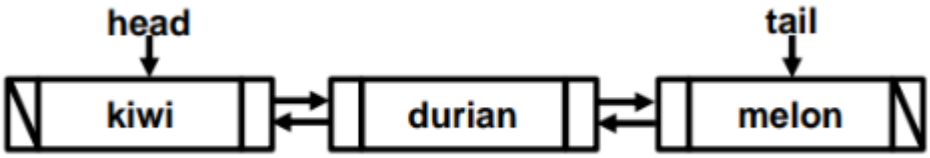
    150 % 100 = 50, yang menjadi pembagi adalah 100 (kapasitas hash tablenya) sehingga hash value-nya adalah **50**
  - untuk key 121

    121 % 100 = 21, yang menjadi pembagi adalah 100 (kapasitas hash tablenya) sehingga hash value-nya adalah **21**
  - untuk key 2500

    2500 % 100 = 0, yang menjadi pembagi adalah 100 (kapasitas hash tablenya) sehingga hash value-nya adalah 0.

Note: jika ada collision, perhatikan cara handlingnya.

5. Given a **queue** as follows.



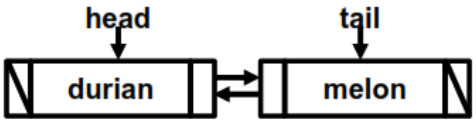
a. **[LO 2; Weight 10%]** Remove one node then add one node that contains mango. Show the process steps and give a short explanation related to the process!

Answer: (jawaban disesuaikan)

Hapus depan

Salah satu caranya misal dengan cara berikut:

```
head=head->next;
free(head->prev);
head->prev=NULL
```

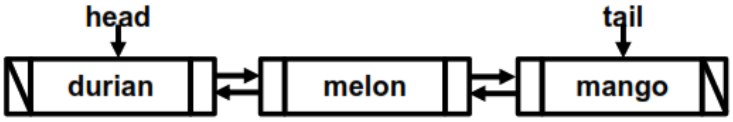


Tambah belakang

Salah satu caranya misal dengan cara berikut:

membuat node baru bersama dengan pointernya misalnya curr. *(misal structnya itu ada isi, *next, *prev)*

```
strcpy(curr->isi, "mango");
curr->next = NULL;
curr->prev = tail;
tail->next = curr;
tail=tail->next;
```



b. **[LO 3; Weight 15%]** Using the C programming language, declare a structure that can hold the data and create a function that can add a content of the linked list.

Answer: (jawaban disesuaikan)

```
struct node
{
    int number;
    struct node *prev;
    struct node *next;
}
node;

node *head;
node *tail;

// Adding node to the queue
void enqueue(int value)
{
    node *n = malloc(sizeof(node));
    if (n == NULL)
    {
        return;
    }

    n->number = value;
    n->next = NULL;
    n->prev = NULL;

    if (head == NULL && tail == NULL)
    {
        head = n;
        tail = n;
    }
    else
    {
        n->next = tail;
        tail->prev = n;
        tail = n;
    }
}
```