

BINUS University

Academic Career: <i>Undergraduate / Master / Doctoral *)</i>		Class Program: <i>International/Regular/Smart Program/Global Class*)</i>	
<input type="checkbox"/> Mid Exam <input checked="" type="checkbox"/> Final Exam <input type="checkbox"/> Short Term Exam <input type="checkbox"/> Others Exam : _____		Term : Odd /Even/ Short *)	
<input checked="" type="checkbox"/> Kemanggisan <input checked="" type="checkbox"/> Alam Sutera <input checked="" type="checkbox"/> Bekasi <input type="checkbox"/> Senayan <input type="checkbox"/> Bandung <input type="checkbox"/> Malang		Academic Year : 2021 / 2022	
Faculty / Dept. : School of Computer Science		Deadline	Day / : Monday / Jul 4 th 2022 Date Time : 13.00 – 16.20 (200 Minutes)
Code - Course : COMP6048 – Data Structures COMP6048001 – Data Structures COMP6048016 – Data Structures COMP6048049 – Data Structures		Class : All Classes	
Lecturer : Team		Exam Type : Online	
*) Strikethrough the unnecessary items			
<b style="color: red;">The penalty for CHEATING is DROP OUT!!!			

EXAM INSTRUCTIONS

1. For Essay (Number 1 and 2). The answers must be written by **hand on paper and only used ballpoint or using digital pen**. You **cannot submit the screenshot from any simulation**.
2. Convert your essay answers (1,2,3) in 1 PDF file using this format: **NIM.pdf**
3. For case problem: The submission code is in .cpp file and using this format: **NIM.cpp**
4. All your answers either essay (NIM.pdf) or case (NIM.cpp) should be zipped and submitted through <https://exam.apps.binus.ac.id/>. Other than that, the submission won't be accepted for any reasons. (Note : Please zip both files using this format: **NIM.zip**)
5. Make sure your file size **is not more** than <https://exam.apps.binus.ac.id/> standard
6. The exam will be marked as 0 if any **plagiarism** is found

Learning Outcomes

- LO1:** Explain the concept of data structures and its usage in Computer Science
- LO2:** Illustrate any learned data structures and its usage in application
- LO3:** Apply data structures using C

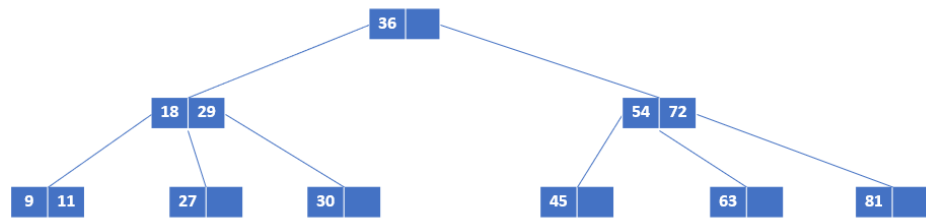
Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022

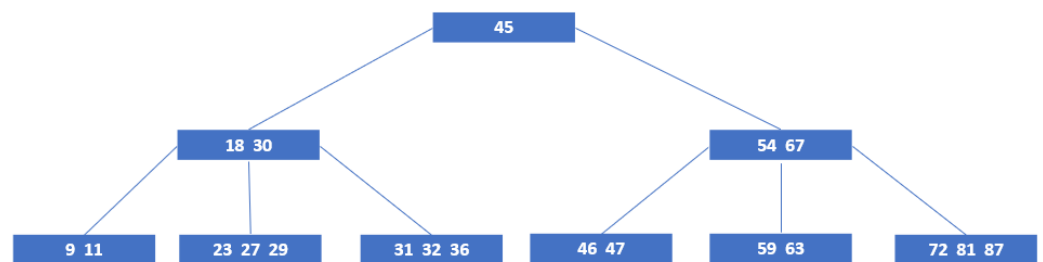
I. Essay (60%)

1. B-Tree

- a. [LO1, LO2, 10 points] Consider 2-3 Tree given below. Insert 67, 87, 32, 23, 31 in the tree

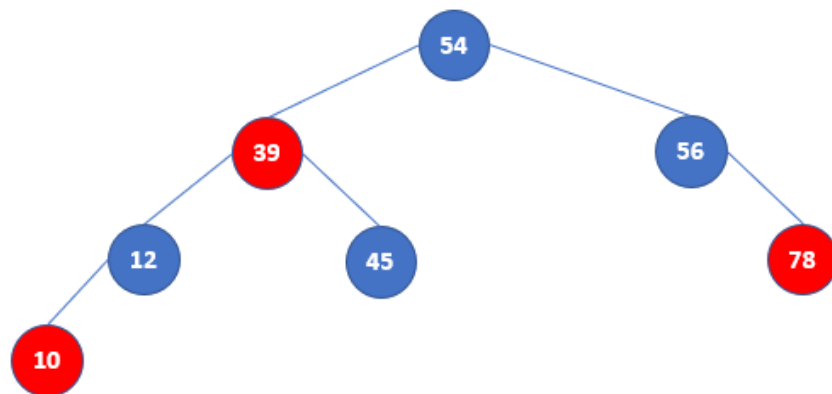


- b. [LO1, LO2, 10 points] Consider B-Tree Order 5 given below. Delete 29, 27, 67, 45 from the tree. Use right most child of left subtree approach if needed



2. Red Black Tree

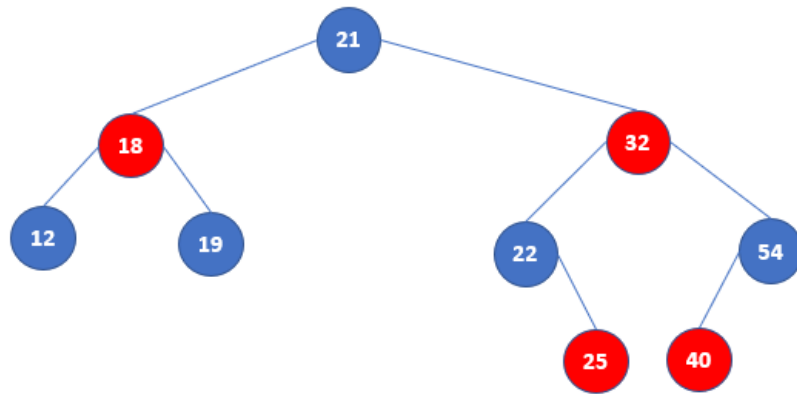
- a. [LO1, LO2, 10 points] Consider Red Black Tree given below. Insert 34, 67, 32, 40 in the tree



- b. [LO1, LO2, 10 points] Consider Red Black Tree given below. Delete 25, 22, 12 from the tree.

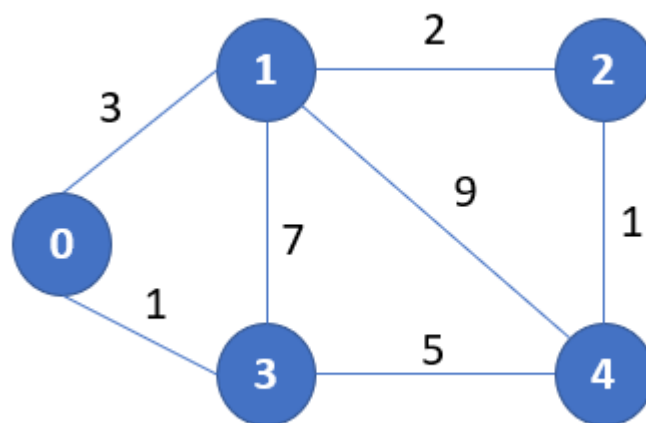
Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022



3. [LO1, LO2, LO3, 20 points] Snippet (MST)

Consider Graph given below to complete the code



```

#include<stdio.h>
#include<limits.h>
#define MAX 50
#define N 5 //number of Nodes
#define S 0 //starting vertex

void fixMatrix(int adjList[MAX][MAX]){
    for(int i=0;i<N;i++){
        for(int j=0;j<N;j++){
            if(adjList[i][j]==0)adjList[i][j]=INT_MAX;
        }
    }
}

void findMST(int start, int adjList[MAX][MAX], int mst[MAX][MAX])
{
    int visited[MAX],d[MAX],parent[MAX];
    int min,u,v;

    for(int i=0;i<N;i++){
        d[i]=adjList[start][i];
        visited[i]=0;
        parent[i]=start;
    }
}
  
```

Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022

```

visited[start]=1;

int k=0;
for(int i=0;i<N-1;i++){

    min=INT_MAX;

    //a. complete this line to select edge that has minimum weight (10)

    visited[u]=1;

    mst[k][0]=parent[u];
    mst[k][1]=u;
    k++;

    for(v=0;v<N;v++)
        if(visited[v]==0 && (adjList[u][v]<d[v])){
            d[v]=adjList[u][v];
            parent[v]=u;
        }
}

}

int main(){

    int adjList[MAX][MAX]={
        //b. complete this line to Initialize adjacency matrix for Graph given (5)
    };

    int mst[MAX][MAX];
    fixMatrix(adjList);
    findMST(S,adjList,mst);

    printf("Edges of MST:\n\n");
    printf("Edge => Weight\n");
    printf("=====\n");

    int cost=0;
    for(int i=0;i<N-1;i++){
        int v1=mst[i][0];
        int v2=mst[i][1];
        printf("%2d%2d => %d\n",v1,v2,adjList[v1][v2]);

        //c. complete this line to compute total cost of MST(5)
    }
    printf("=====\n");
    printf("Total Cost: %d\n",cost);

    return 0;
}

```

Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022

```
Edges of MST:

Edge => Weight
=====
0 3 => 1
0 1 => 3
1 2 => 2
2 4 => 1
=====
Total Cost: 7
```

II. Case (40%)

[LO1, LO2, LO3, 40 points] Café Jolly Soul is a place that sells trendy drinks. The drinks they sell and the drink codes are : **Boba Coffee (BC)**, **Moccapucino (MC)**, **Pandan Coffee (PC)**, **Boba Thai (BT)**, **Choco Pandan (CP)** and **Cheese Choco (CC)**. Cafe Jolly Soul has a program to input order and view sales:

- A. First Line, User will input **T** orders
- B. For the next **T** lines, in each line, User will input the following format: **N [Drink Code]**
N is the order quantity
- C. Output will display 2 Top Sales with the following output format:
 1. **[Drink Names] [Sales Amount]**
 2. **[Drink Names] [Sales Amount]**

Try implementing Heap into the **Café Jolly Soul** program using C

CONSTRAINTS

$1 \leq T \leq 100$

$1 \leq N \leq 100$

Drink Codes: **Boba Coffee (BC)**, **Moccapucino (MC)**, **Pandan Coffee (PC)**, **Boba Thai (BT)**, **Choco Pandan (CP)** and **Cheese Choco (CC)**

Sample 1:

```
5
10 BC
5 MC
3 CC
2 BC
21 BT
Top Sales:
1. Boba Thai 21
2. Boba Coffee 12
```

Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022

Sample 2:

```
10
5 BC
3 MC
13 CC
12 BC
2 BT
10 PC
2 PC
4 CP
5 MC
25 MC
Top Sales:
1. Moccapucino 33
2. Boba Coffee 17
```

-- Selamat Mengerjakan --

Verified by,

Hidayaturrahman (D6423) and sent to Program on June 16, 2022