Estd.1995

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

POGIL

Process Oriented Guided Inquiry Learning (**POGIL**) is a student-centered, group-learning instructional strategy and philosophy developed through research on how students learn best. POGIL was devised in 1994 to better teach general chemistry.

About POGIL:

There are two crucial aspects to the design of a POGIL activity. First, sufficient appropriate information must be provided for the initial "Exploration" so that students are able to develop the desired concepts. Second, the guiding questions must be sequenced in a carefully constructed manner so that not only do students reach the appropriate conclusion, but at the same time various process and learning skills are implemented and developed.

Typically the first few questions build on students' prior knowledge and direct attention to the information provided by the model. This is followed by questions designed to help promote the recognitions of relationships and patterns in the data, leading toward some concept development. The final questions may involve applying the concepts to new situations and generalizing students' new knowledge and understanding. Thus, POGIL activities follow the structure of the learning cycle of exploration, concept invention and application, and has a strong basis in constructivism.

In contrast to traditional classrooms, students in a POGIL classroom work in small groups (of 3 or 4) on a specially designed activity. Each student is assigned a role, such as manager, recorder, spokesperson or reflector. The instructor serves as a facilitator who listens to the discussion and intervenes at appropriate times to guide student learning. In groups, students discuss the answers to carefully crafted questions that lead them to consider the general ideas in question and to construct their own understanding of important course concepts. As ideas are formulated, groups share their findings and understanding to new and increasingly difficult problems or contexts.

Rather than having the instructor begin class by defining terms and laying out concepts, students work actively to master material and formulate a deeper understanding of content. Built into the experience is the support of a variety of important process skills, including communication, teamwork, and critical thinking, which translates to a more complete understanding of the entire concept, and a lasting understanding of the material.

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

Estd.1995

1. POGIL TASK ON - SORTING

Department of CSE
II/IV B.Tech I Semester

INDEX

Faculty Information:

Y. Bhanusree, Assistant Professor, VNRVJIET, Hyderabad.

Learning Objectives

After completing this activity, learners should be able to:

- Understand and visualize that there will be numerous algorithms/programs for a problem
- Understand and identify different strategies of sorting.
- Able to evaluate/ calculate the complexities of the algorithms.
- Estimate and identify the best possible algorithm for a problem in terms of efficiency.
- Identify and use appropriate asymptotic notations

- Should be able to know the best, worst and average cases for an algorithm.

Prerequisites
Before starting this activity**,** learners should have an experience, of writing pseudocode.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software, so teams can see each other's work easily.

Activity Notes

- The facilitator should spend 5minutes for introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.

Activity History

1/10/14 Prepared by Y.Bhanusree bhanusree_y@vnrvjiet.in

Before you start, complete the form below to assign a role to each member.

If you have 3 people, combine Manager & Reflector.

Team		**Date**	
Team Role		**Team Member**	
Recorder: records all answers & questions, and provides copies to team & faculty.			
Speaker: talks to faculty and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Other:			

Introduction

Sorting is the basic operating any used in every form of application. Even if you take the contact lists in the cell phone or arrange icons on the desktop in an order/ save files in a folder the sorting algorithm is executed in the background. Let's find out the roots of it in this POGIL sheet.

(10 min) CASE 1 Planning of strategy:

Given a bowl of marbles arrange them in the order of their size.

1: Which marble did you select for the first time?

2: Which marble did you select second time?

3: How did you select the first marble Describe in sentence?

(5 min) CASE 1 Identifying strategy:

Write down the strategy of arranging the marbles in order.

(10 min) CASE 2 Planning of strategy:

Given the play cards one by one arrange them in the sequence.

1. Note the method (each sequence of steps) of arranging them in the order.

(5 min) CASE 2 Identifying strategy:

Write down the strategy of arranging them in order.

(10 min) CASE 3 Planning of strategy:

Provided the access for any two objects only among 5 at a time. Arrange them in an order .

(5 min) CASE 2 Identifying strategy:

Write down the strategy of arranging them in order.

(15 min) Identifying and comparing the techniques

1. Name the basic methods observed in case1, case2, case3.You provide a name based on the technique you have worked for it.

2. Provided 10 objects in each case list number of steps which method do you consider requires less number of steps by a human.

3. Is this the same number of steps for the computer also?

(30 min) Tracing with values and finding complexities

1. Given the elements

25 41 21 14 37 18 20 7 235 35

Trace the number of steps using all the three basic sorting techniques

2. Find equations for each method to show number of steps in sorting and derive its asymptotic notation.

3. Compare the three techniques based on the number of steps/ Asymptotic notation.

(15 min) Coding

1. Write code for swapping procedure.

2. For exchange sort write the snippet of code to identify requirement of swapping & code of swapping.

```
If (condition)
{

    Code for swap;

}
```

3. Write code for selecting the smallest value.

4. For selection sort write the snippet of code to swap the smallest value with the tracing element.

5. Write snippet of code for implementing insertion sort.

Estd.1995

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

2. POGIL TASK ON – SEARCHING

Department of CSE

Learning Objectives

After completing this activity, learners should be able to:

- Understand and visualize that there will be numerous algorithms/programs for a problem
- Able to evaluate/ calculate the complexities of the algorithms.
- Estimate and identify the best possible algorithm for a problem in terms of efficiency.
- Identify and use appropriate asymptotic notations
- Should be able to know the best, worst and average cases for an algorithm.

Prerequisites

Before starting this activity, learners should have an experience, of writing pseudocode.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software, so teams can see each other's work easily.

Before you start, complete the form below to assign a role to each member.

If you have 3 people, combine Manager & Reflector.

Recorder: records all answers & questions, and provides copies to team & faculty.	
Speaker: talks to faculty and other teams.	

Manager: keeps track of time and makes sure everyone contributes appropriately.	
Other:	

Introduction

In computing, we often must search in a set for a particular item. As computer scientists, we are particularly interested in searching very large sets, with thousands or millions of values. For example, the Harvard University Library has roughly 16,000,000 volumes, and the US Library of Congress has roughly 22 million cataloged books, and over 100,000,000 total items. In this activity, we use a simple game to explore some basic searching algorithms. This will also help us explore more general concepts in algorithm design and analysis, so studying searching is useful even though very few of us may need to implement searching algorithms, since efficient techniques are part of most software libraries.

Hi-Lo Game

Hi-Lo is a number guessing game with simple rules.
a. There are two players – A and B.
b. Player A thinks of a number from 1 to 100.
c. Player B guesses a number.
d. Player A responds with "too high", "too low", or "you win".
e. Players B and A continue to guess & respond until B wins (or gives up).

I. (10 min) Player Strategies

1. (3 min) Play the game a few times to ensure that everyone understands the rules.

2. (2 min) List up to 3 ways to clarify the rules.

3. (3 min) Describe 4-5 different strategies that Player B could use to guess numbers.

Try to have a mixture of simple and clever strategies.
Name each strategy and list it in the first column of the worksheet.

Before you continue, review progress with the facilitator.

II. (10 min) Comparing strategies

1. (2 min) Evaluate each strategy with regard to how **quickly** it will find the right answer, by rank ordering from 1 (least guesses) to 5 (most guesses).
Add the rankings to the worksheet in a column labeled **Quick**.

2. (2 min) Evaluate each strategy with regard to how **easy** it is to describe or specify, by rank ordering from 1 (easiest) to 5 (hardest).
(Suppose you had to explain each strategy to a first-grader so that she could play the game.) Add the ranking to the worksheet in a column labeled **Easy**.

3. (1 min) For each strategy, multiply the quick rank by the easy rank, and add the product to the worksheet in a column labeled **Product**.

4. (3 min) In complete sentences, describe the relationships between the two sets of rankings.

Before you continue, review progress with the facilitator

III. (10 min) Worst & Average Case Performance

1. (2 min) Discuss and list the pros & cons of measuring program speed with a stopwatch.

2. (3 min) For each strategy, determine the **worst case** (maximum) number of guesses required to win.

Add the numbers to the worksheet in a column labeled **Worst**.

3. (3 min) For each strategy, determine the **average case** (typical) number of guesses required to win.
Add the numbers to the worksheet in a column labeled **Average**.

Note that the **minimum** number of guesses is always 1 – it's nice to be lucky.

4. (2 min) List 3 reasons why it would be useful to have more precise, quantitative ways to measure and discuss the speed of an algorithm.

Before you continue, review progress with the facilitator.

IV. (10 min) Effect of Input Size

1. (3 min) Assume that Player A chooses a number from 1 to 1000.
For each strategy, what are the worst case & average case number of guesses?
Add the numbers to the worksheet in columns labeled "1K Worst" and "1K Average".

2. (4 min) **Optional**: Assume that Player A chooses a number from 1 to N. (For example, N=100, N=1000, N=1,000,000)
For each strategy, what are the worst case & average case number of guesses in terms of N?
Add the expressions to the worksheet in columns labeled "N Worst" and "N Average".
(Hint: you've already done N=100 and N=1000; consider other values before generalizing to N.)

3. (3 min) Describe the pros & cons of analysing performance in terms of input size N

WORKSHEET

Strategy name	Quick	Easy	prod	Worst	Average	1k Worst	1k Average	N worst	N Average

Estd.1995

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

3. POGIL TASK ON – LINEAR DATA STRUCTURE

Department of CSE

INDEX

Faculty Information:

Y. Bhanusree, Assistant Professor, VNRVJIET

Learning Objectives

After completing this activity, learners should be able to:

- Discover the necessity of a data structure named LINEAR LIST, LINKED LIST, STACK ans QUEUE and identify their properties.
- Identify the operations and variables required for implementing.
- Exemplify these data structures to show their operations.
- Identify/Predict the applications .
- Implement using C++ generic concepts

Prerequisites

Before starting this activity, learners should have an experience, of writing ADT.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software,so teams can see each other's work easily.

Activity Notes

- The facilitator should spend 5minutes for introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.

Activity History

20/1/13 Prepared by Y. Bhanusree bhanusree6912@gmail.com

Revised and restructured on 25/9/14 by Bhanusree.Y Bhanusree_y@gmail.com

Before you start, complete the form below to assign a role to each member.

Team NO		**Date**	
Team Roles		**Team Member** **(Write your Roll-number)**	
Recorder: records all answers & questions, and provides copies to team & faculty.			
Speaker: talks to faculty and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Other:			

MODEL

Consider all these different ways of storing objects/data

a. Stack of files

b. Books Storage shelf

C.Shuttlecock barrel

d. Treasure hunt

Spend 5 min in discussing the differences in the way of storing and retrieval.

Part- I [5 min]

Consider the above systems and name them.

a.

b.

c.

d.

Discuss with your facilitator before you proceed further.

Part- II [5 min]

If for a data structure **Natural numbers** the

Instances are { 1, 2, 3, x,....,y,.....Max}

Operations are Add( ), Sub( ), Mul( ), Div( )... [**But not Area(), Perimeter().]**

1. Write Instances and operations of Linear list

2. Write Instances and operations of Linked list

3. Write Instances and operations of Stack.

4. Write Instances and operations of Queue.

Part- III [10 min]

If this is the ADT of a natural number:

Abstract Data Type Natural Numbers

{

Instances:

Ordered collection from 0 to Max.

Operations:

Add(): Return sum of x and y x,yϵ NaturalNumbers and sum<Max;

Sub(): Return diff of x and y x,yϵ NaturalNumbers and diff>0;

.

.

.

}

1. Write ADT of Linear List

2. Write ADT of Linked List

3. Write ADT of Stack

4. Write ADT of Queue

Part- IV [10 min]

Linear lists Implementation:

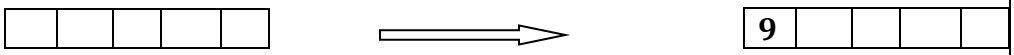
Can Include the functions like Create(),Isempty(),Size(),get (index), Indexof(), erase(index), Insert(), Display().

Write the class for Linearlist with constructor to initialize the index to '0'

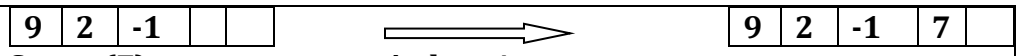
```
#define MAX 50
template<class T>
class Linearlist
 {
  Protected:
        T list[MAX];
           int element, index;
   public:
     Linearlist()


                                  //write body of constructor
        }

      int isempty()
      {                          //declare and define the member functions



       }

      int isfull()
       {                         //declare and define the member functions



       }
   Void .....();
     .
     .
     .




    };
```

Create():

Ask user for an element and then place it in the array[index].



Create(9) index=1



Create(7) **index=4**

```
template<class T >
void Linearlist<T>::create()
{

                    //Check if max is reached or not.
                          // if not then place the element

}
```

Insert():

Ask user for an element and then insert it in the index provided by user itself.

```
templ.....
void .....::insert()
{

                    //check for is full() and then ask the
                     position and element

                    // if pos==index
```

// if pos>max

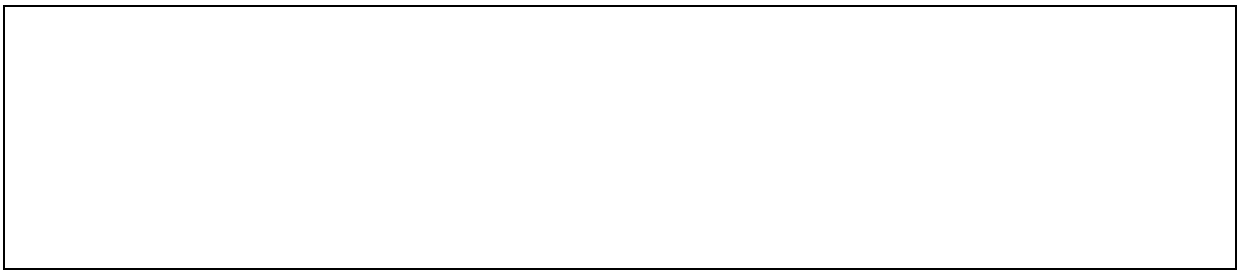
//if pos in between

}

Delete(): // empty, last position, middle positon

Search(): **// empty, non empty conditions.**

Display(): **// empty and not empty**



VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

Estd.1995

4. POGIL TASK ON – QUEUES

Department of CSE

INDEX

Faculty Information:

Y. Bhanusree, Assistant Professor, VNRVJIET

Learning Objectives

After completing this activity, learners should be able to:

- Discover the necessity of a data structure named LINEAR LIST, LINKED LIST, STACK ans QUEUE and identify their properties.
- Identify the operations and variables required for implementing.
- Exemplify these data structures to show their operations.
- Identify/Predict the applications .
- Implement using C++ generic concepts

Prerequisites

Before starting this activity, learners should have an experience, of writing ADT.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software,so teams can see each other's work easily.

Activity Notes

- The facilitator should spend 5minutes for introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.

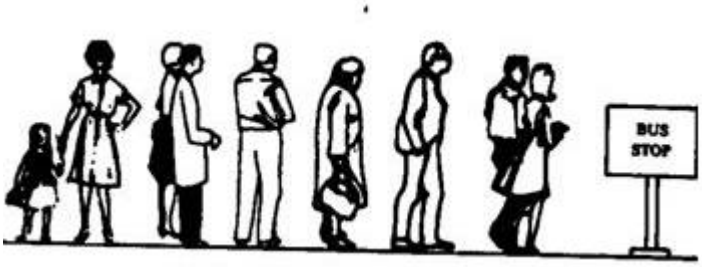
Activity History

14/10/14 Prepared by Y. Bhanusree bhanusree6912@gmail.com

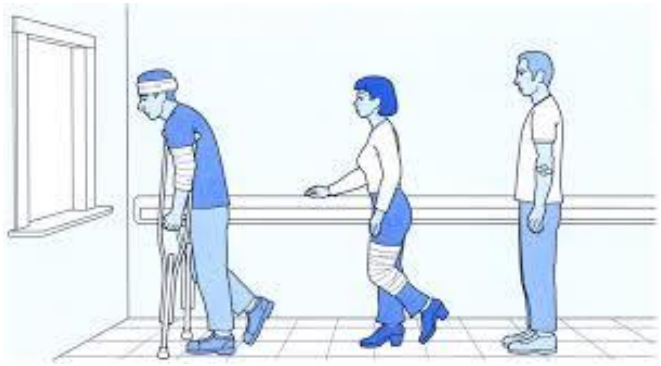
Before you start, complete the form below to assign a role to each member.

Team NO		**Date**	
Team Roles		**Team Member** **(Write your Roll-number)**	
Recorder: records all answers & questions, and provides copies to team & faculty.			
Speaker: talks to faculty and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Other:			

PRIORITY QUEUE:



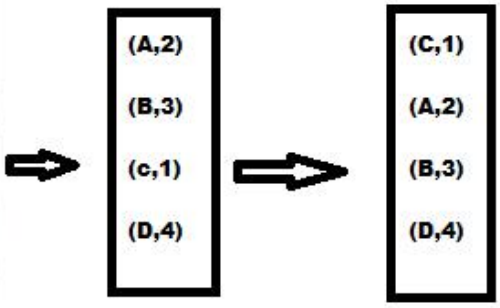
Is this the proper order to enter bus?(Think of the mother and child at 6th position and old lady at 3rd position)



What is the order of admitting in to hospital? Justify?

Know what should be the order?

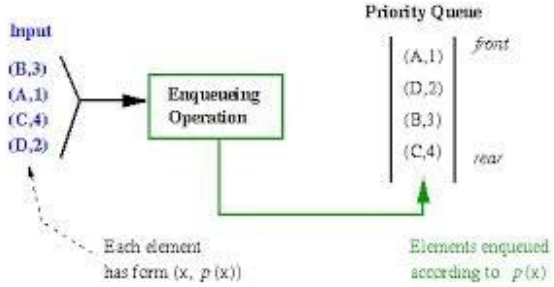
Lets know convert the situation in to a mathematical form



A(2nd) B(3rd) C (Ist) D(4rth)

Name this? **Name this?**

Finally the model can be visualized as



Part- I [5 min]

Consider the above systems and name them.

ADT and CLASS IN C++, definitions, working wth lists, time complexity, and then heaps, definition ,properties and operations

a.

b.

c.

d.

Discuss with your facilitator before you proceed further.

Part- II [5 min]

If for a data structure **Natural numbers** the

Instances are { 1, 2, 3, x,....,y,.....Max}

Operations are Add( ), Sub( ), Mul( ), Div( )... [**But not Area(), Perimeter().]**

1. Write Instances and operations of Linear list

2. Write Instances and operations of Linked list

3. Write Instances and operations of Stack.

4. Write Instances and operations of Queue.

Part- III [10 min]

If this is the ADT of a natural number:

AbstractDataType NaturalNumbers
{
Instances:
Ordered collection from 0 to Max.
Operations:
Add(): Return sum of x and y x,yϵ NaturalNumbers and sum<Max;
Sub(): Return diff of x and y x,yϵ NaturalNumbers and diff>0;
.
.
.
}

1. Write ADT of Linear List

2. Write ADT of Linked List

3. Write ADT of Stack

4. Write ADT of Queue

Part- IV [10 min]

Linear lists Implementation:

Can Include the functions like Create(),Isempty(),Size(),get (index), Indexof(), erase(index), Insert(), Display().

Write the class for Linearlist with constructor to initialize the index to '0'

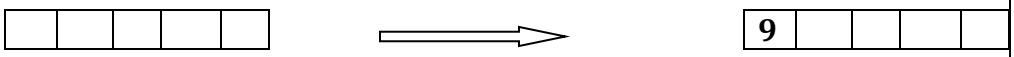
```
#define MAX 50
template<class T>
class Linearlist
 {
  Protected:
        T list[MAX];
           int element, index;
   public:
     Linearlist()


                                  //write body of constructor
        }

      int isempty()
      {                          //declare and define the member functions



       }

      int isfull()
      {                          //declare and define the member functions



       }
   Void .....();
     .
     .
```

.

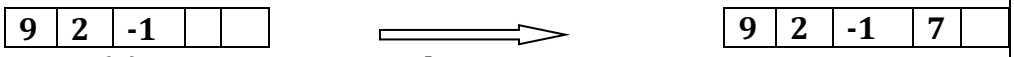
};

Create():

Ask user for an element and then place it in the array[index].



Create(9) **index=1**



Create(7) **index=4**

```
template<class T >
void Linearlist<T>::create()
{

                    //Check if max is reached or not.
                        // if not then place the element

}
```

Insert():

Ask user for an element and then insert it in the index provided by user itself.

```
templ.....
void .....::insert()
{
```

//check for is full() and then ask the position and element

// if pos==index

// if pos>max

//if pos in between

}

Delete(): // empty, last position, middle positon

Search(): **// empty, non empty conditions.**

Display(): **// empty and not empty**

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

Estd.1995

5. POGIL TASK ON – DICTIONARIES

Department of CSE

INDEX

Faculty Information:

Y. Bhanusree, Assistant Professor, VNRVJIET

Learning Objectives

After completing this activity, learners should be able to:

Identify the use of dictionaries, Implement dictionaries using Linear lists and Hashing, Compare complexities, benefits, problems of different forms of dictionaries.

`

Prerequisites

Before starting this activity, learners should have an experience, of writing ADT, searching, little about time complexity.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software, so teams can see each other's work easily.

Activity Notes

- The facilitator should spend 5minutes for introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.

Activity History

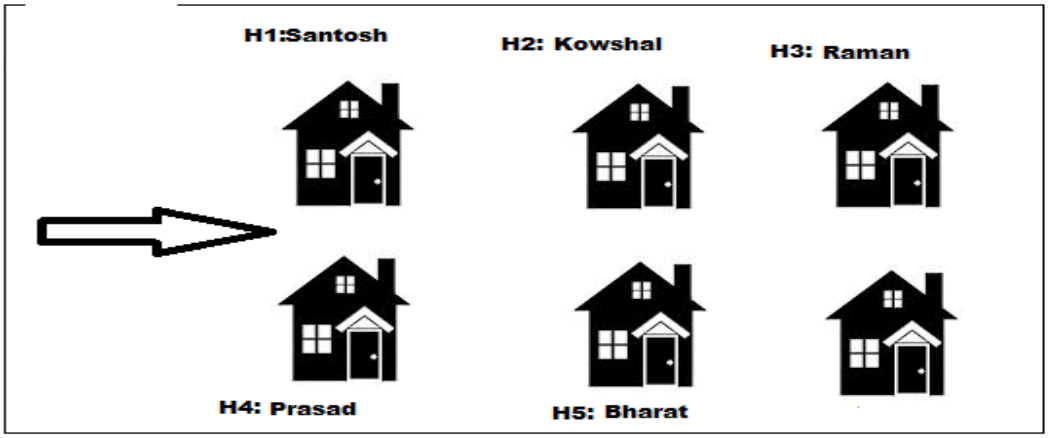
15/10/14 Prepared by Y. Bhanusree bhanusree6912@gmail.com

Before you start, complete the form below to assign a role to each member.

Team NO		**Date**	
Team Roles		**Team Member** **(Write your Roll-number)**	
Recorder: records all answers & questions, and provides copies to team & faculty.			
Speaker: talks to faculty and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Other:			

Santosh, Kowshal, Raman, Prasad, Bharat have constructed a new colony with 6 houses. Consider the following cases.

Case I: Allotment of houses is in the sequence as follows

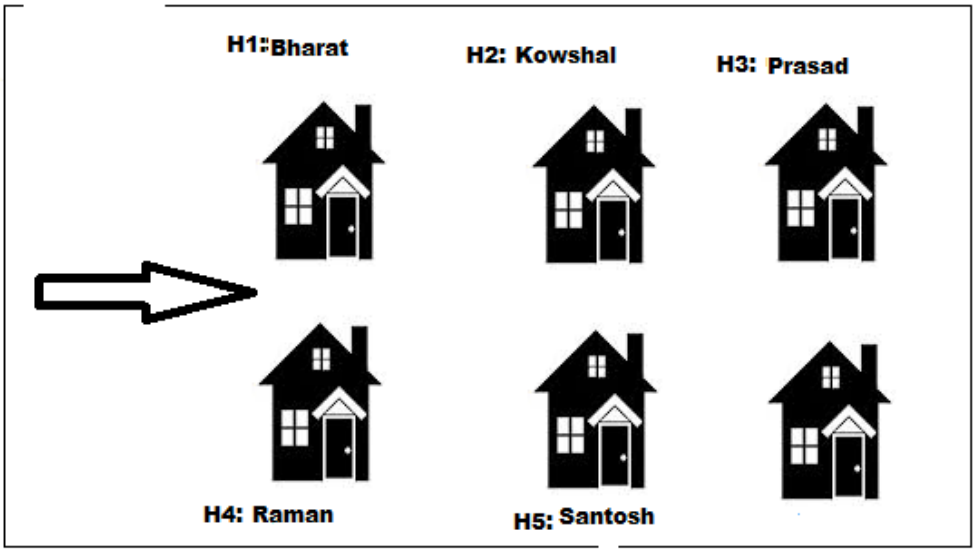


Starting from H1
count in how many times you can find the house of

- **Bharat ----------**
- **Santosh----------**
- **Prasad----------**

Chandu wants to occupy the vacant home. Count in how many times he can find a vacant home----------

Case 2: Allotment has been done in alphabetical order as follows



Starting from H1 you know the list of people and that they are in order.
count in how many times you can find the house of

- **Bharat ----------**

- **Santosh----------**
- **Prasad----------**

Chandu wants to occupy the appropriate vacant home. Count in how many times he can find a vacant home----------

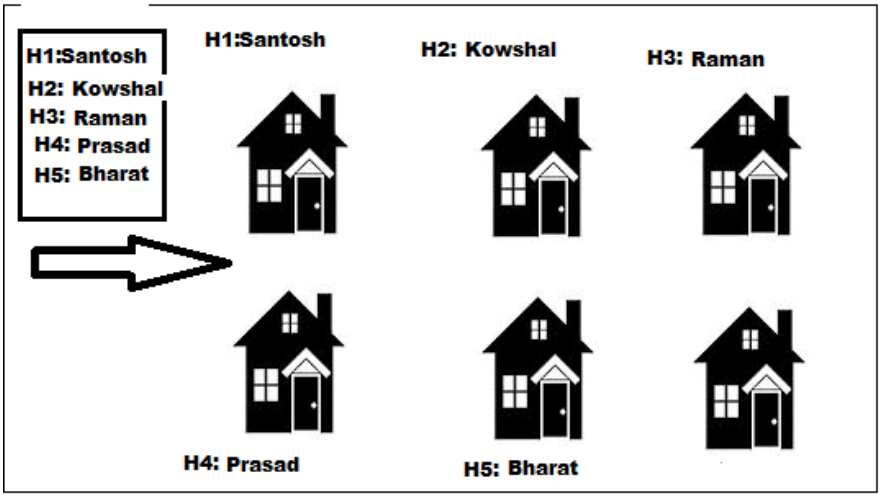
Case 3: For either of the above cases you can ask the House no of the specific person in the first/some house and directly go to the desired house.

(Some times they may not know the exact House number)
Starting from H1
count in how many times you can find the house of

- **Bharat ----------**
- **Santosh----------**
- **Prasad----------**

Chandu wants to occupy the appropriate vacant home. Count in how many times he can find a vacant home----------

Case 4: An address board is kept in the starting for reference



count in how many times you can find the house of

- **Bharat ----------**
- **Santosh----------**
- **Prasad----------**

Chandu wants to occupy the appropriate vacant home. Count in how many times he can find a vacant home----------

Will these be same if the houses are in alphabetical order?

Fill the count from the above case in the following table

Case:	Search	Insert	Erase
Case 1:			
Case 2:			
Case 3:			
Case 4:			

Comparing the size of houses to be 'n' write the variable for number of times

Data structure name	Case	Search	Insert	Erase
	Case 1			
	Case 2			
	Case 3			
	Case 4			

So let us learn the concept of implementing _____.

Part- I [5 min]

To implement searching using dictionaries you need _____and _____.
Definition:

Suggested Operations for dictionaries:

Discuss with your facilitator before you proceed further.

Part- II [15 min]

1. Can a dictionary have two values with same key or vice versa? Discuss and try to get to a conclusion and write justification with 3 examples. (Dictionaries with duplicates)

Justification:

Examples:

2. Should these dictionaries be in sequential or non sequential? Justify with examples and appropriate data structure for implementation.

Sequential **NON Sequential**

Examples:

Data structure:

Part- III [10 min]

AbstractDataType DICTIONARY
{

Instances:

Operations:

}

List the different ways of implementing Dictionaries:

Data structure type	Array/linked list	Time complexity for search	Advantage	Disadvantage

Part- IV [15 min]

1. To implement dictionary with linear list is it better to have a sorted one or unsorted one? Justify

2. Write the code for searching in the array implementation of linear list for dictionary.(Hint: You need to search with key and retrieve the value).

3. Implement the same with linked list implementation

Part- V [15 min]

Skip List: (Consider the Case3 in our model)

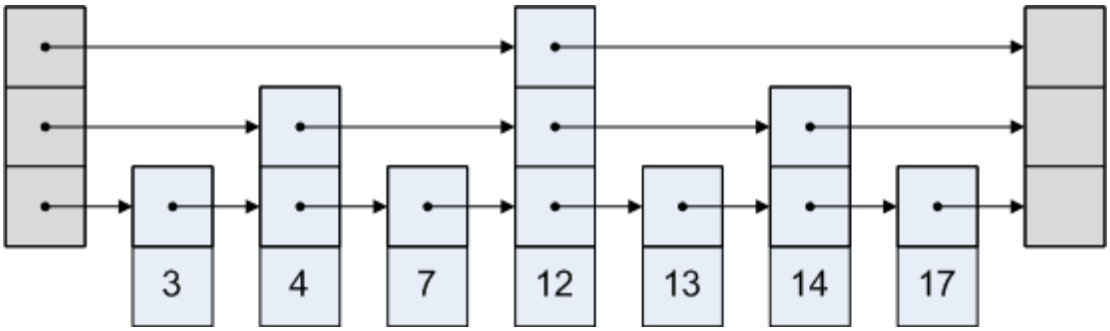
1. Is this way of traversal linear/ non linear. If so what is the expected time complexity for searching and insertion?

2. Construct a sorted linked list (Compulsion for skip list) with 5 elements --- 12, 4, 7, 13, 17, 3, 14.

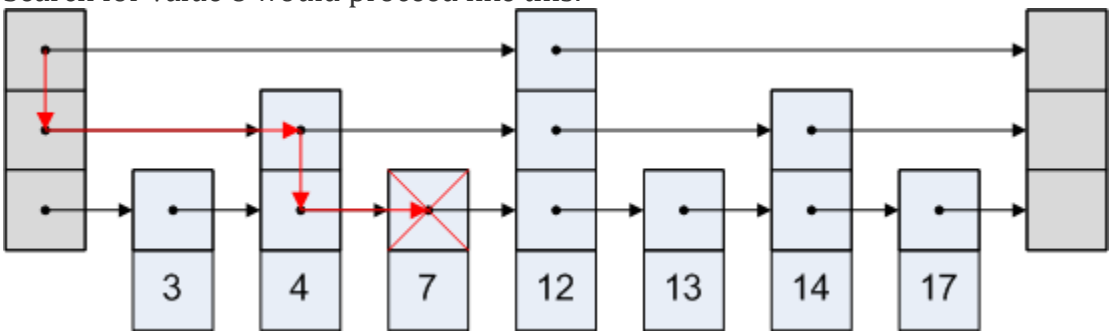
Check how many comparisons do you need to check the value of 12.(Also write time complexity.

3. Reconstruct the sorted linked list to a 2 level list such that there will be a direct link from first node 3 to node 12 along with link from 3 to 4.

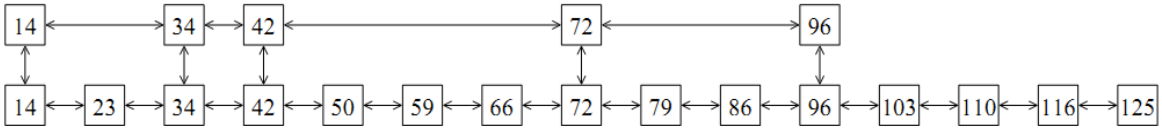
4.Please consider the discussion with the facilitator and redraw the multi level skip list.



Search for value 8 would proceed like this:



14, 23,34,42,50,59,66,72,79,86,96,103,110,116,125




```
public class SkipListEntry
 {
   public String key;
   public Integer value;

   public SkipListEntry up;      // up link
   public SkipListEntry down;    // down link
   public SkipListEntry left;    // left link
   public SkipListEntry right;   // right link

   ...
   (methods)
 }
```

-
```
1.  typedef struct skip_list{
2.          SkipListNode header;
3.          int level;
4.          int MaxLevel;
5.       } *SkipList;
6.
```

p = head;

Search
{

Move to the right until your right neighbor node
contains a key that is greater than k

if (not lowest level)
Drop down one level
else
exit
}

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

Estd.1995

6. POGIL TASK ON - HEAPS

Department of CSE

INDEX

Faculty Information:

Y. Bhanusree, Assistant Professor, VNRVJIET

Learning Objectives

After completing this activity, learners should be able to:

- Discover the necessity of a data structure named Priority queue.
- Identify the operations and variables required for implementing.
- Exemplify this data structure to show their operations.
- Identify/Predict the applications.
- Implement using C++ generic concepts

Prerequisites

Before starting this activity, learners should have an experience, of writing ADT.

Preparation

Optional: Provide the worksheet on the board, a poster, or in presentation software,so teams can see each other's work easily.

Activity Notes

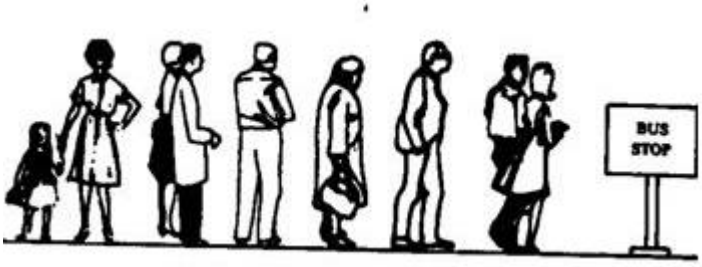
- The facilitator should spend 5minutes for introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.

Activity History

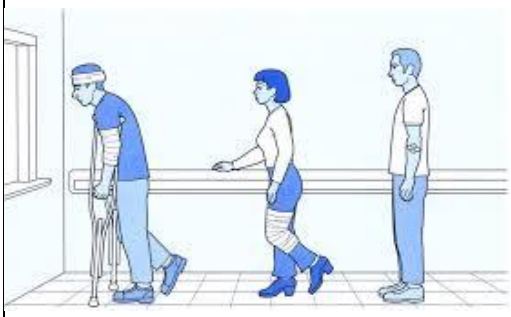
14/10/14 Prepared by Y. Bhanusree bhanusree6912@gmail.com

Before you start, complete the form below to assign a role to each member.

Team NO		**Date**	
Team Roles		**Team Member** **(Write your Roll-number)**	
Recorder: records all answers & questions, and provides copies to team & faculty.			
Speaker: talks to faculty and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Other:			



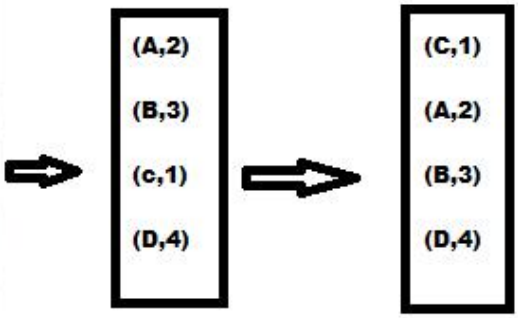
Is this the proper order to enter bus?(Think of the mother and child at 6th position and old lady at 3rd position)



What is the order of admitting in to hospital? Justify?

Know what should be the order? **(DON'T see the next picture in the model)**

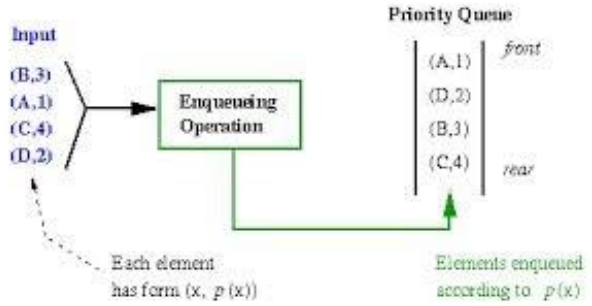
Let's know convert the situation in to a mathematical form



A(2nd) B(3rd) C (Ist) D(4rth)

Name this? **Name this?**

Finally the model can be visualized as



Part- I [15 min]

1. **Consider the above systems and name them.**

2. **Define the data structure.**

3. **Identify some applications of this data structure**

4. **Can you implement this data structure with linear list and linear linked list**

5. **Can you name some disadvantages in implementation of this data structure with linear list(LL) and linear linked list(LLL).(After you answer this you should be able to choose either of LL, LLL for implementing.**

6. **Write ADT and CLASS IN C++.**

Discuss with your facilitator before you proceed further.

Part- II [10 min]

1. **Suggest techniques which are best suited for sorting.**

2. **What modifications can be done to it, to have a maximum or minimum number with time complexity=O(1).**

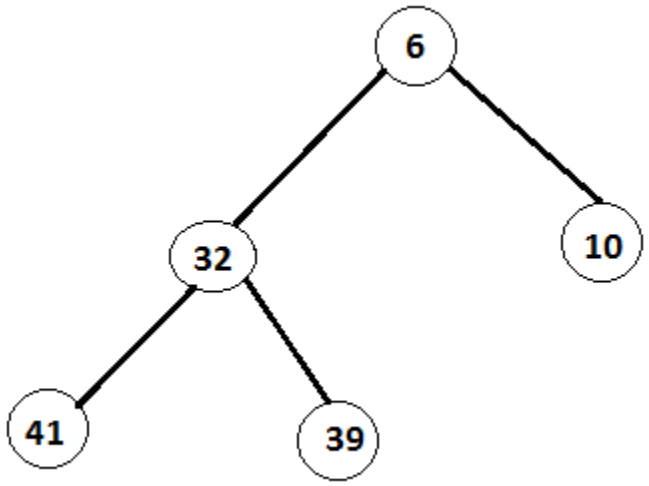
   Draw a linear and non linear ds and identify the time complexity.
   BST not suitable then what
   Max and min tree and Max and min heap

Part- III [30 min]

Consider the tree in the diagram and mention, what type of tree it is?

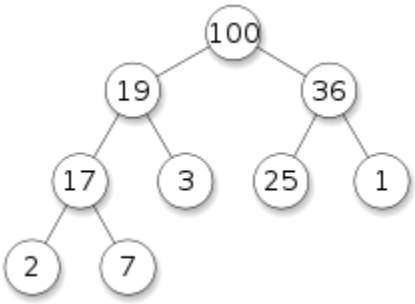
a.

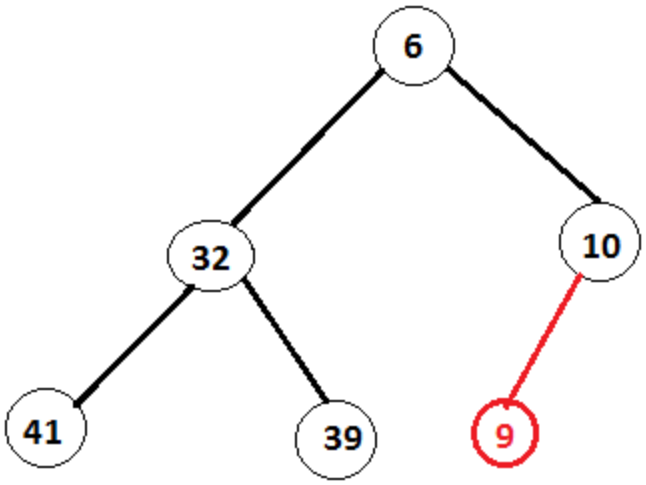
b.



1. **Name the heap.**

2. **Know let us insert a value 9 in to the Min heap.**





Is it a min heap know?

3. **Suggest a technique to make it a min heap?**

4. **Know let us insert 45 in to min heap.**

   Is it a min heap?

5. **Insert 5 in to min heap.**

6. **Consolidate the whole process of insertion.**

7. **Consolidate the whole process of making it a min heap.**

LAB POGIL

<u>Model: Take inputs from the class discussion</u>

1. Considering the priority as having the highest value, construct the **complete binary tree for the following data**

20 12 35 15 10 80 30 17 2 1

a. Heapify Max using percolate down principle.

b. Insert 13 and heapify

c. Delete Max.

2. Considering the priority as having the highest value construct the **complete binary tree for the following data**

3 5 6 7 20 8 2 9 12 15 30 17

a. Heapify Max using percolate down principle.

b. Insert 15, 20, 45 and heapify

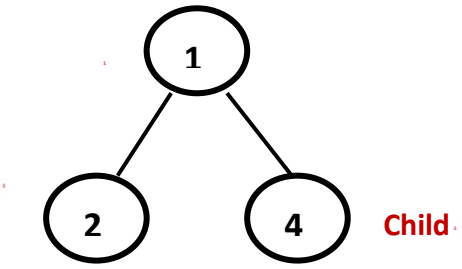
c. Delete Max.

d. Using delete max no of times prepare a sorted list.

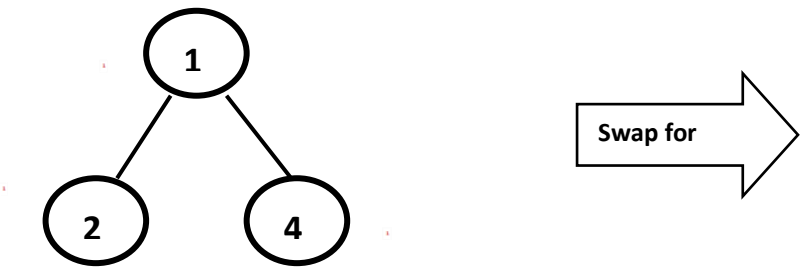
Code Generation:

I. Heap Initialization/Percolate down:

1. If the last leaf node is called with position **child** what is the parent position?



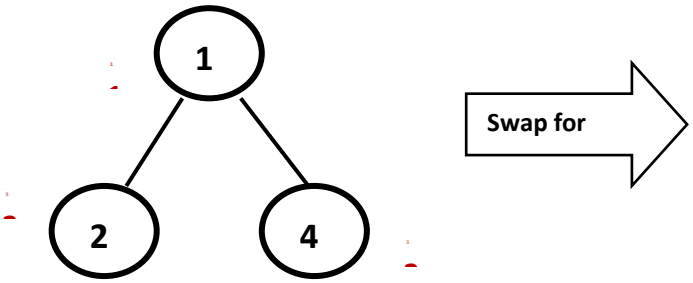
2.



Why did you select 42 instead of 22 for moving to root?

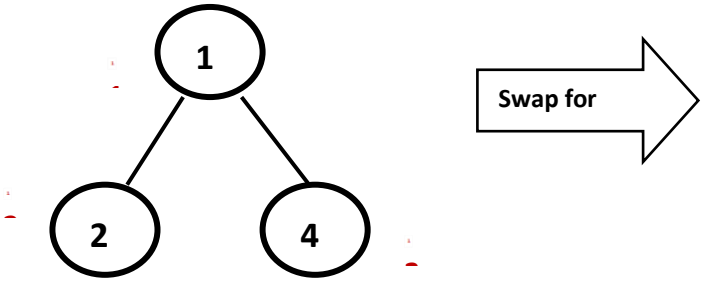
Write the code for selecting the max(child,child+1) for heapification process.

3.



Write the code snippet for heapification in this case.

4.



Write the code snippet for heapification in this case.

5. Write the logic for Percolate down or heap initialization using the above cases.

II. Deletion

STEPS **CODE**

a. removing the root element----------

b. Move the last leaf element in the heap to root-----

c. Decrement the size----------

d. Percolate down----------

III. Insertion

STEPS **CODE**

a. Accept element from user at the hole position---

b .Increment the size----------

c. percolate up(send parent element to hole)

until element is bigger than parent-----

d. copy element to hole whent the correct position

identified----------

IV. Heap SORT

STEPS **CODE**

a. Delete max element for maximum

size times and copy to sorted array----------