Name: ______________________________

# *16.unified*
# *Introduction to Computers and Programming*

## *SOLUTIONS to Examination*

4/30/04
9:05am - 10:00am

Prof. I. Kristina Lundqvist
Spring 2004

Grading Section:

| | |
|---|---|
| Question 1 (5) | |
| Question 2 (15) | |
| Question 3 (10) | |
| Question 4 (35) | |
| Question 5 (10) | |
| Question 6 (15) | |
| Question 7 (10) | |
| ***Total 100*** | |

You have 55 minutes to take this examination. Do not begin until you are instructed to do so. This is a closed book examination. No external materials are permitted, including calculators or other electronic devices. All answers must be written in the examination paper. This examination consists of 7 questions and 12 pages (not including this cover page). Count the number of pages in the examination paper before beginning and immediately report any discrepancy to the invigilator. Should you need to do so, you may continue your answers on the back of pages.

**Do not forget to write your name on each page**.

Name: ______________________________

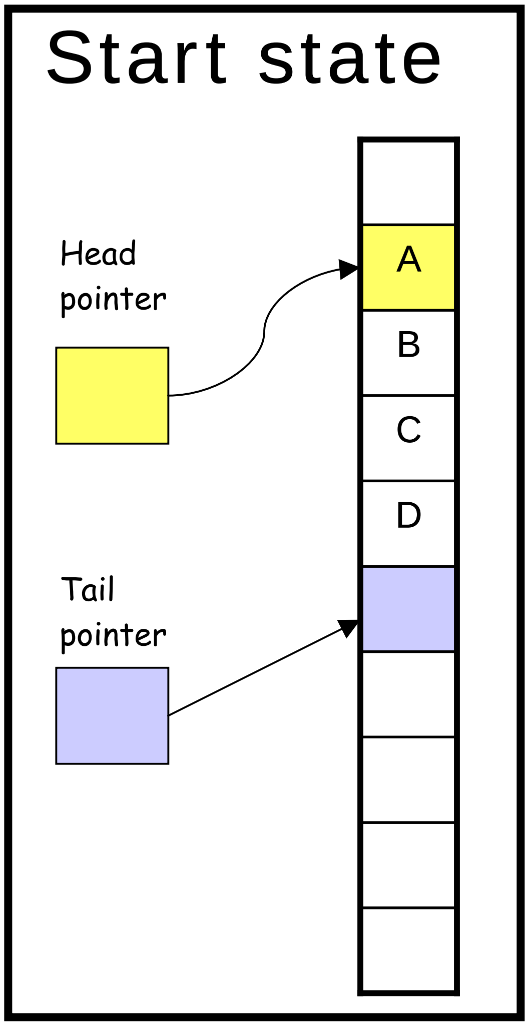
**Problem 1 - Queue** **(5 points)**

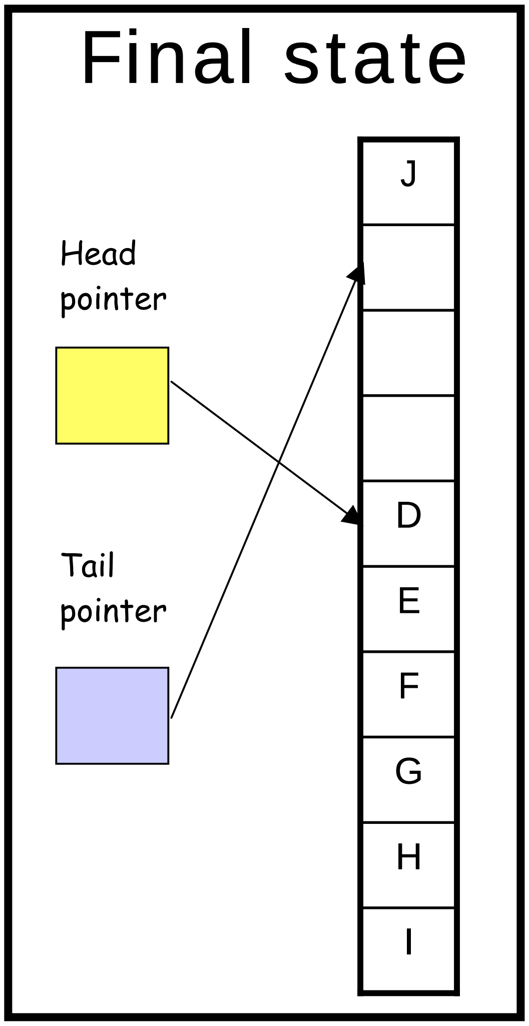
Consider the circular queue of size 10, as shown in the Figure below (Start state). The circular queue contains letters A through D.

Assume the following 9 operations take place in sequence

1. Insert entry E
2. Insert entry F
3. Remove one entry
4. Remove one entry
5. Insert entry G
6. Insert entry H
7. Remove one entry
8. Insert entry I
9. Insert entry J

Show the contents of the circular queue after (Final state) performing all of the operations. Where are the head and tail pointers located?



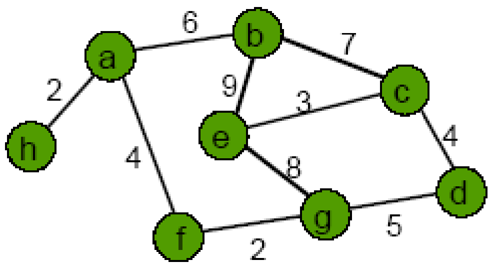


Name: ________________________________

**Problem 2 – MST** **(15 points)**

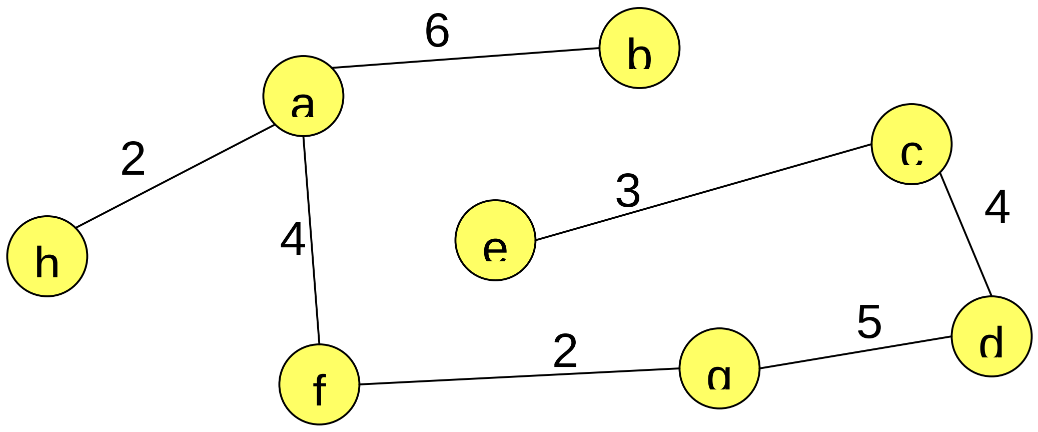
Using the given graph below, what is the 'minimum-weight spanning tree'? Which algorithm are you using? Show clearly, step by step, on the next page, how the algorithm is used. Finally, draw the resulting Minimum-weight spanning tree here below.

Given Graph:

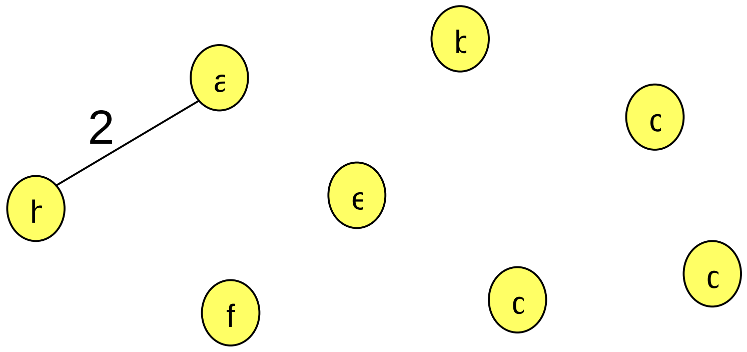


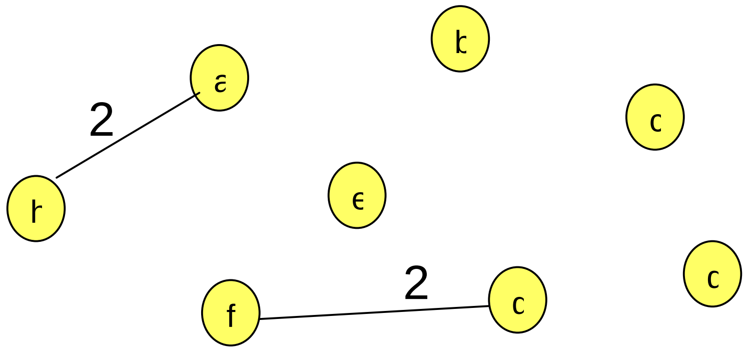
Name of algorithm you are using: Kruskal's algorithm

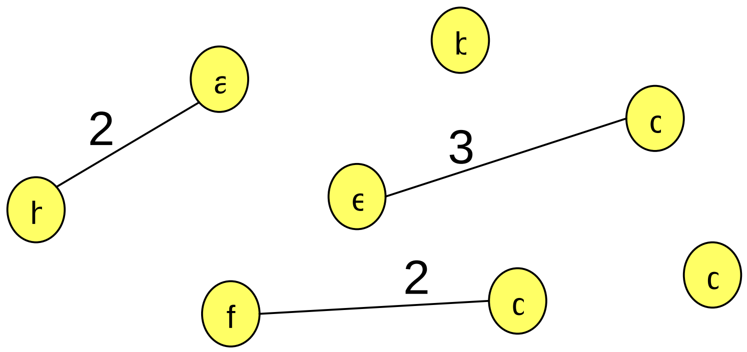
Minimum-weight spanning tree:

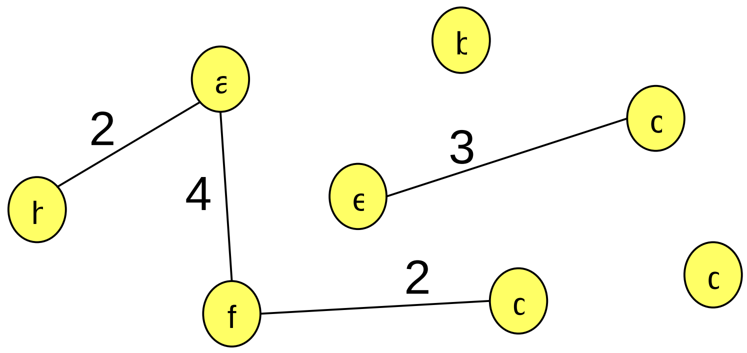


Name: ______________________________

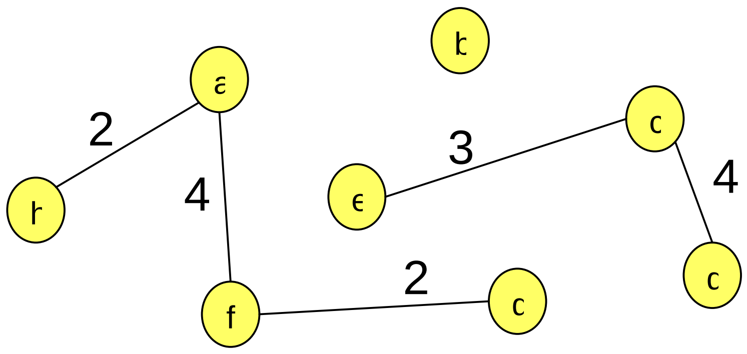


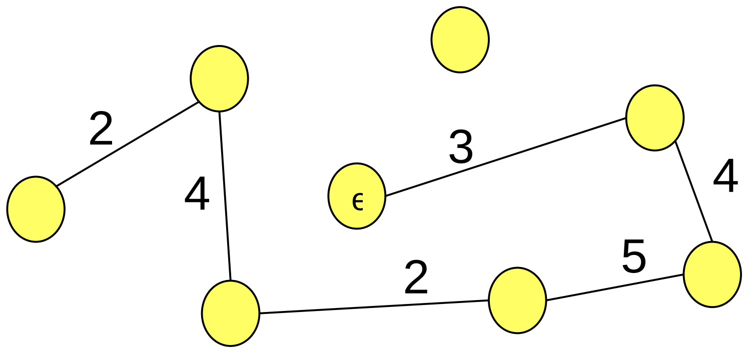


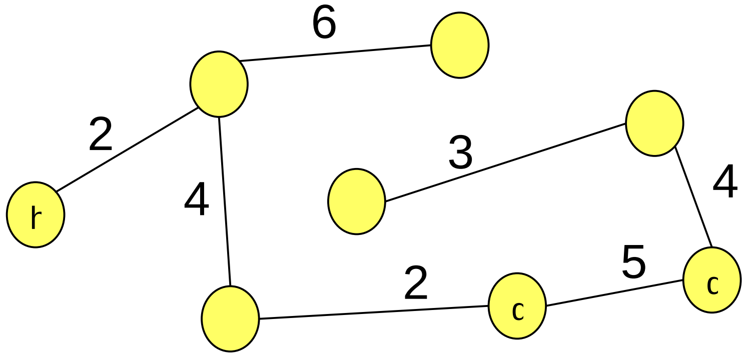




Name: ___________________________________







Name: ______________________________

## Problem 3 – Big-O (10 points)

**Part a)** Show the computation of T(n) and the Big-O complexity for the code shown below. **(6 points)**

```
Statement                                                    Work

function Compare (Value1, Value2 : Float )
                              return Float is
   Result : Float;                                           c1
   Minimum : Float := 2.0;                                   c2
begin -- Compare

   if Value1 < Value2 then                                   c3

      Result := Value1;                                      c4

   else                                                      c5

      Result := Value2;                                      c4

   end if;                                                   c7

   return Minimum;                                           c8

end Compare;
```

T(n) = c1 + c2 + c3 + 2c4 + c5 + c6 + c7 + c8 = **c**

O(n) = **O(1)**

**Part b)** What is the result passed out by the program when the input values are 10.8 and 10.2? **(4 Points)**

**The output is 2.0 because of the "`return Minimum;`" statement.**

Name: ________________________________

**Problem 4 – Ada Tree Implementation** **(35 points)**

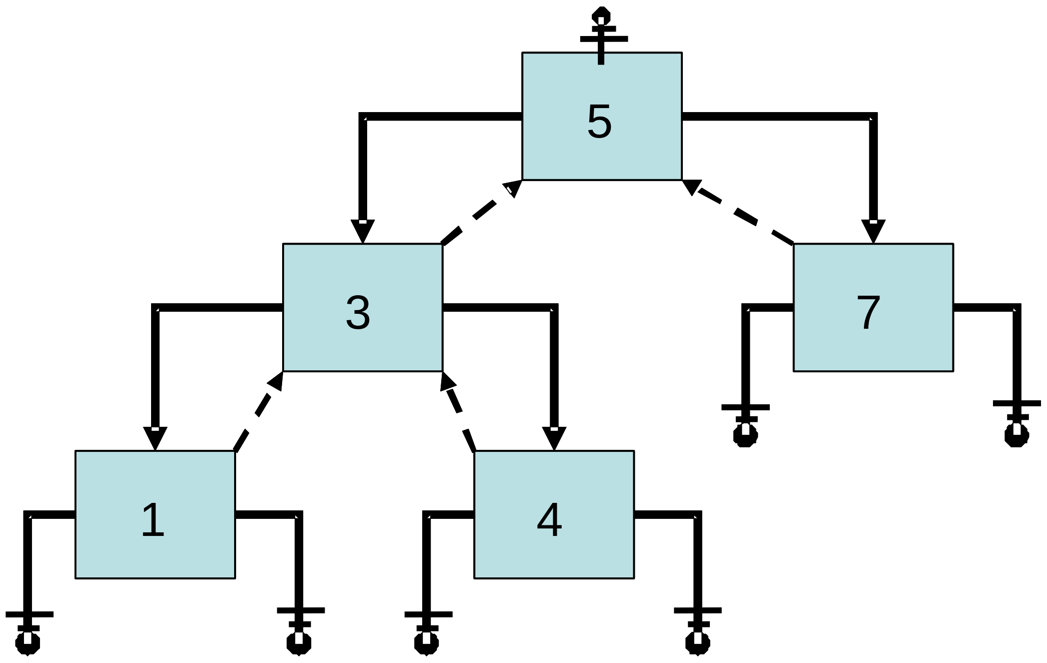


Figure 3. Ordered Tree

**Part a.** Define the Ada95 record declaration for the node in the tree shown in Figure 3.
**(3 points)**

Note: The dotted line links to the parent node, while the solid lines link to the children (maximum of two).

```
type Node;
type Node_ptr is access Node;
type Node is
  record
               Element     : Element_type;
               Left_Child  : Node_ptr;
               Right_Child : Node_ptr;
               Parent : Node_ptr;

  end Record;
```

Name: ________________________________

**Part b.** Write an algorithm to perform ordered insertion into the ordered tree shown in Figure 3. **(18 points)**

***Algorithm***

```
Create two node pointers Temp, New_Node.
Allocate memory for New_Node using new.

Initialize the fields of New_Node
    Element is set to input element
    All pointers are set to null

If Root = null
    Set root to New_Node

Else
    Set Temp to Root.
    Set Flag to False

    While Flag is False

        If Temp.Element is smaller than Element

            If Temp has no right child
                Temp.right_child := New_Node
                New_Node.Parent := Temp
                Set Flag to True
            Else
                Move Temp to Temp.right_child

        Else

            If Temp has no left child
                Temp.left_child := New_Node
                New_Node.Parent := Temp
                Set Flag to True
            Else
                Move Temp to Temp.left_child
```

Name: ________________________________

**Part c.** Implement your algorithm as an Ada95 procedure, which accepts the root node, and the element to be inserted, and performs the ordered insertion operation. **(10 points)**

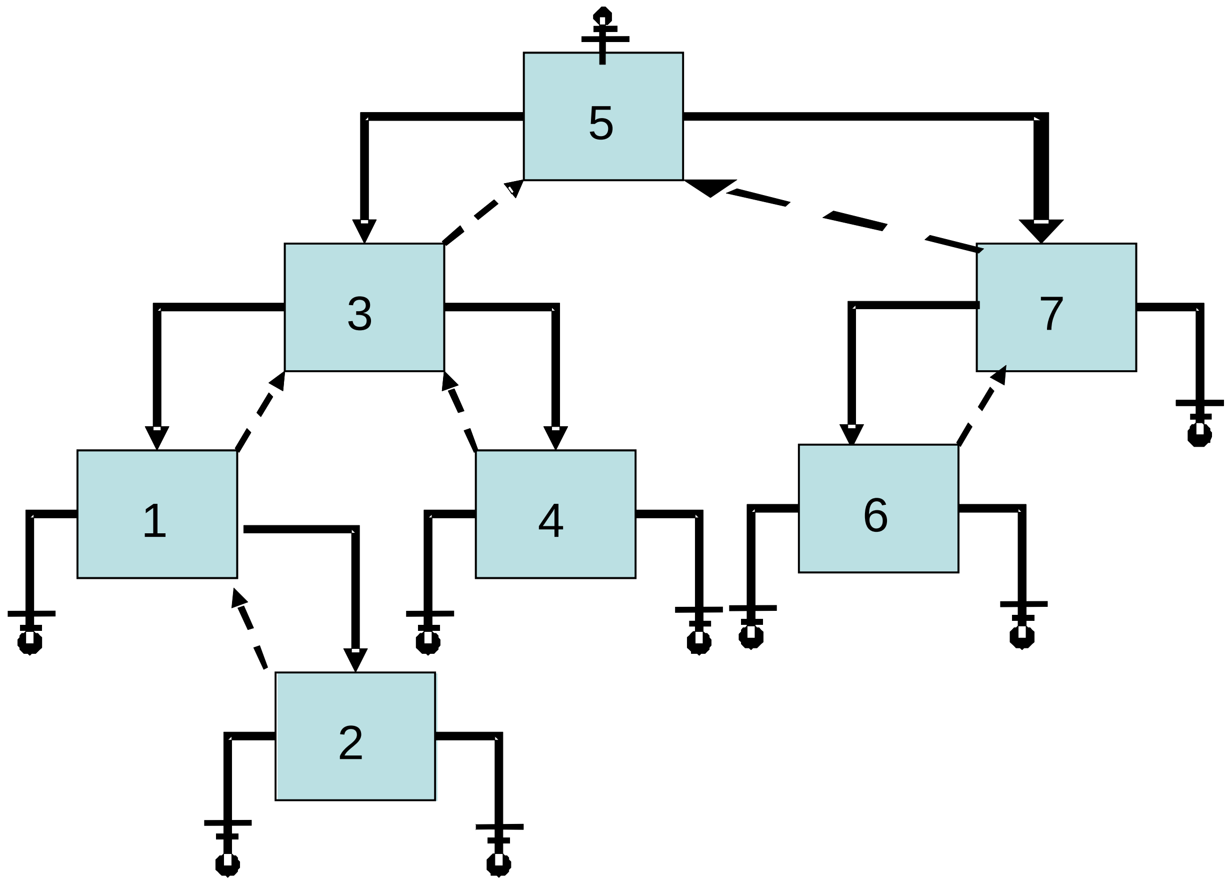
**Program Code**

```
procedure Insert (Root    : in out Node_ptr;
                  Element : in     Element_Type) is

   Temp, New_Node : Node_ptr;
   Inserted : Boolean;

begin

   New_Node:= new Node;
   New_Node.Element := Element;
   New_Node.Left_Child := null;
   New_Node.Right_Child := null;
   New_Node.Parent := null;

   if Root = null then
      Root := New_Node;
   else
      Inserted := False;
      Temp := Root;
      loop
         exit when Inserted = True;

         if Temp.Element < Element then
            if Temp.Right_Child /= null then
       Temp:= Temp.Right_Child;
            else
               Temp.Right_Child:= New_Node;
               New_Node.Parent := Temp;
        Inserted := True;
            end if;
         else
            if Temp.Left_Child/= null then
               Temp := Temp.Left_Child;
            else
               Temp.Left_Child:= New_Node;
               New_Node.Parent := Temp;
                Inserted := True;
            end if;
         end if;
      end loop;
   end if;
end Insert;
```

Name: ________________________________

**Part d.** Update the tree shown below, after inserting the elements ‘2’ and ‘6’ using your algorithm. Show all the requisite links (including nulls) in the diagram.

**(4 points)**



Name: ________________________________

## Problem 5 – Ada Exception Handling (10 points)

```
 1. with Ada.Text_Io, Ada.Integer_Text_Io, Ada.Unchecked_Deallocation;
 4.
 5. procedure Demo_Robust_Programming is
 6.
 7.   subtype My_Integer is Integer;
 8.   type My_Integer_Ptr is access all My_Integer;
 9.
10.   My_Num     : My_Integer;
11.   My_Num_Ptr : My_Integer_Ptr;
12.
13.   procedure Free is
14.   new Ada.Unchecked_Deallocation(My_Integer, My_Integer_Ptr);
15.
16. begin
17.   My_Num_Ptr := new My_Integer;
18.
19.   Free(My_Num_Ptr);
20.
21.   Ada.Text_Io.Put("Please enter an integer: ");
22.   Ada.Integer_Text_Io.Get(My_Num);
23.
24.   My_Num_Ptr.All := My_Num;
25.
26.   Ada.Text_Io.Put(Integer'Image(My_Num_Ptr.All));
27.
28. exception
29.   when Constraint_Error =>
30.     Ada.Text_Io.Put_Line("Unsafe Pointer Handling");
31.   when Ada.Text_Io.Data_Error =>
32.     Ada.Text_Io.Put_Line("Trying to enter a non integer value");
33.
34. end Demo_Robust_Programming;
```

**Part a)** What is the program behavior when the user enters a floating point number? Justify your answer. **(4 points)**

The program will generate a **constraint error** on Line 24 because the user is trying to access already deallocated memory

**Part b)** What is the program behavior when the user enters a valid integer and Line 24 is commented out? Justify your answer. **(6 points)**

When line 24 is commented out, the **constraint error is raised** in line 26 for exactly the same reason as before: the memory has been deallocated.

Name: ________________________________

## Problem 6 – Asymptotic Complexity – Divide and Conquer (15 points)

What is the Big-O complexity of the algorithm shown below? Detail the steps in computing T(n) and O(n).

```
3Sort(A, left, right)

if (left < right)                                          c1

    first_split := (left + right) / 3                      c2

    second_split := (first_split + right)/2                c2

    3Sort(A, left, first_split)                        T(n/3)

    3Sort(A, first_split+1, second_split)              T(n/3)

    3Sort(A, second_split+1, right)                    T(n/3)

    Merge(A, left, first_split, second_split, right)     O(n)
```

Therefore $T(n) = 3T(n/3) + 2c2+c1 + O(n)$
$= 3T(n/3) + O(n) + C$

Therefore correlating to the simplified master theorem:

$cn^k = O(n) \rightarrow k = 1$

$aT(n/b) = 3T(n/3)$

Therefore:

$T(n) = O(n^k \log_b n)= O(n^1 \log_3 n) = O(n \log_3 n)$

Name: ________________________________

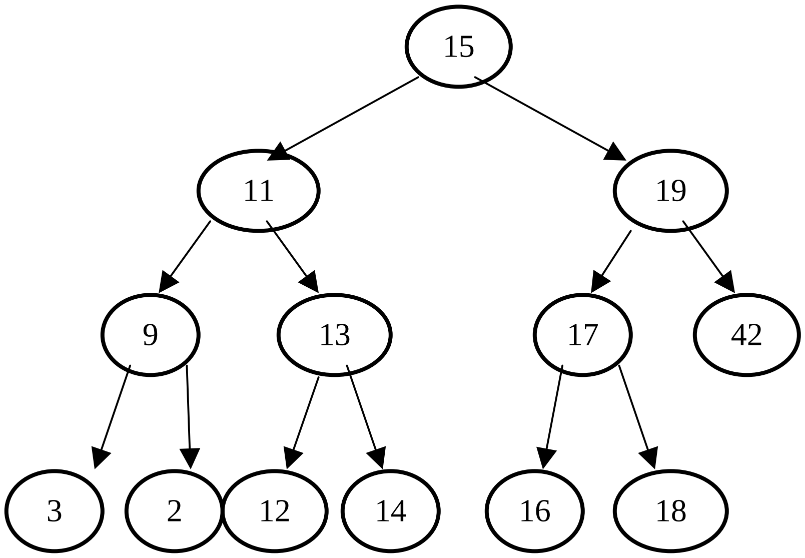
## Problem 7 (10 points)

Multiple Choice Questions. For each question, select the correct answer from the choices, and **write the chosen letter in the box provided** next to each question.

**Answer**

1. The tree below is a:
   a. Full binary tree
   b. Sorted binary tree
   c. Heap

A



2. Take a look at the Tree above, which of the following statements is correct?

   a. Vertex 13 is at Level 3
   b. The height of the tree is 3
   c. The height of the tree is 2

B

3. The postfix notation of a*b+c/d is

   a. ab*cd/+
   b. ab+cd*/
   c. abcd*+/

A

Name: ________________________________

4. When it comes to a stack, which of the following statements is false?

   a. The process of deleting an object is called Pop
   b. All insertions and deletions of elements take place at the same end of the data structure
   c. Stacks are FIFO structures

C

5. I have put my name in the upper right corner on all pages of the quiz

   a. Yes
   b. No
   c. I will do it by 10am today

A/B/C