95-771 Data Structures and Algorithms
Fall 2018

Midterm Exam
Carnegie Mellon University

Name Key

Andrew ID ____________________

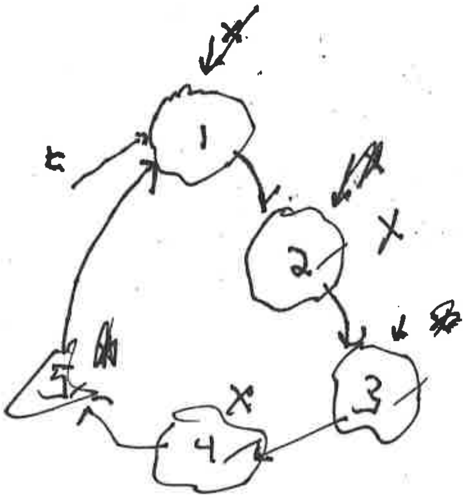
There are 118 points on this exam. Your score will be a percentage of 118.

Part 1. Reading Code with Pointers & Big O (20 points)

You will be asked to show the exact output of the following program in Question 1.

```
// This program goes with question 1
class Node {
   int data;
   Node next;

   public Node(int v) {
      data = v;
   }

  public static void main(String[] args) {

     int n = 5;
     Node t = new Node(1);
     Node x = t;
     for (int i = 2; i <= n; i++) {
        x.next = new Node(i);
        x = x.next;
     }
     x.next = t;
     x = t.next;
     while(x != t) {
        System.out.println(x.data);
        x = x.next;
     }
  }
}
```

N
5



1. Show the exact output of the program above. (6 Points)

2 3 4 5

Note

1 2 3 4 5
-4

2 3 4 5 1
-3

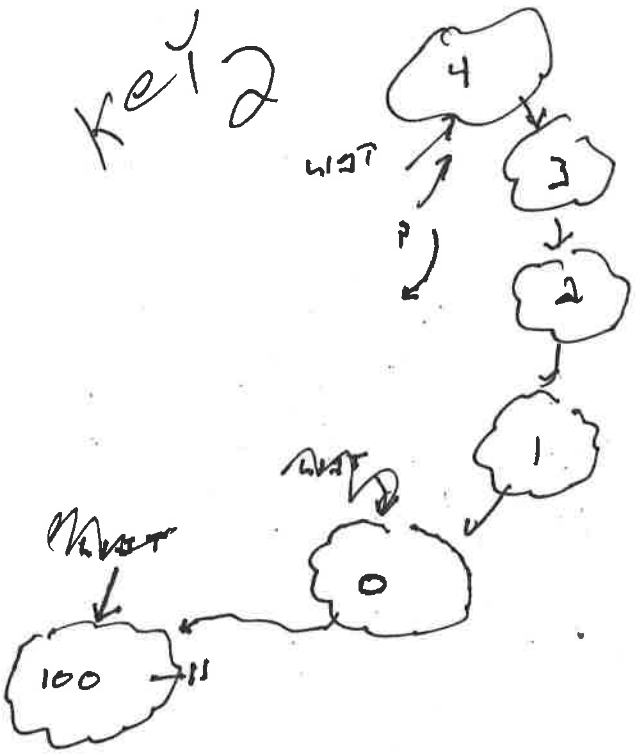
5 4 3 2 1
-5

.0

Study the execution of the following program. Question two appears below.

Key 2

```
// This program goes with question 2.
class Node {
      int data;
      Node next;

      public Node(int v, Node p) {
         data = v;
         next = p;
      }

      public static Node change(Node x) {
         Node t,y,r;
         y = x;
         r = null;
         while(y != null) {
            t = y.next;
            y.next = r;
            r = y;
            y = t;
         }
         return r;
      }

      public static void main(String[] args) {

         Node list = new Node(100,null);
         for(int i = 0; i < 5; i++) {
            list = new Node(i,list);
         }
         Node p = list;
         while (p != null) {
            System.out.print(p.data + "   ");
            p = p.next;
         }
         list = change(list);
         System.out.println();
         p = list;
         while (p != null) {
            System.out.print(p.data + "  ");
            p = p.next;
         }
      }
}
```



2.a There are exactly two lines of output from this program. The first line of output is what? (5 Points)

4 3 2 1 0 100

2.b There are exactly two lines of output from this program. The second line of output is what? (5 Points)

100 0 1 2 3 4

10

Key 3

2.c Consider the routine named “change” in the program in Question 2. Suppose the list is of size n and suppose T(n) is the number of operations performed by the routine named “change”. CIRCLE all of the following that are true statements about T(n). 4 Points.

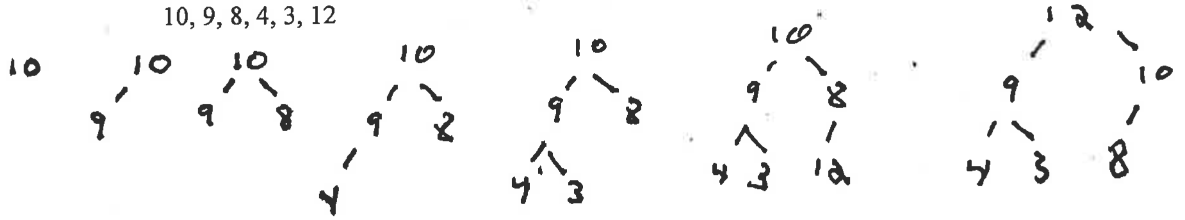
1. T(n) ε O(LogN)
2. T(n) ε O(1)
3. T(n) ε Ω(N^2)
[✓ 4.] T(n) ε O(N)
5. T(n) ε Θ(LogN)
[✓ 6.] T(n) ε O(2^n)
[✓ 7.] T(n) ε Ω(1)
[✓ 8.] T(n) ε Θ(N)

MAX OFF = 4

Part 2. Heaps (12 points)

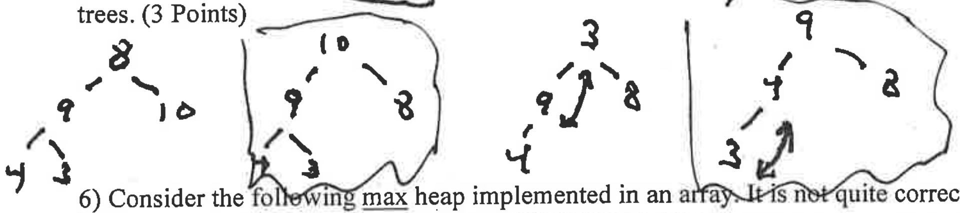
3) Insert the following 8 numbers into a max heap. Draw a new tree for each heap insertion. (4 Points)

10, 9, 8, 4, 3, 12



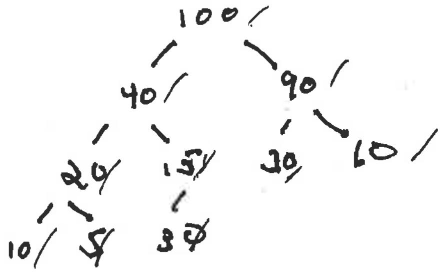
4) What is the height of the tree that you drew in question 3? (2 Points) 2

5) Perform exactly two deleteMax() operations on the heap that you drew in question 3. Draw the resulting trees. (3 Points)



6) Consider the following max heap implemented in an array. It is not quite correct. To make it a proper max heap exactly one swap must occur. What two numbers (child and parent) need to be swapped in order to make this a max heap? (3 points). PLACE CHECK MARKS NEXT TO THE TWO NUMBERS THAT NEED TO BE SWAPPED.

100
40
90
20 ✓
15
30
60
10
5
30 ✓



,0

Key Q

Part 3. Binary Trees (16 points)

7. Parts (a), (b), (c) refer to the following binary tree:

```
        13
      /    \
    40       97
   / \      / \
  7   9    2   5
   \      /
    8    3
```

(a) List the data that would be accessed by a pre-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (3 points)

13, 40, 7, 8, 9, 97, 2, 3, 5

(b) List the data that would be accessed by an in-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

7, 8, 40, 9, 13, 3, 2, 97, 5

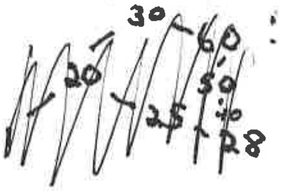
(c) List the data that would be accessed by a level-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

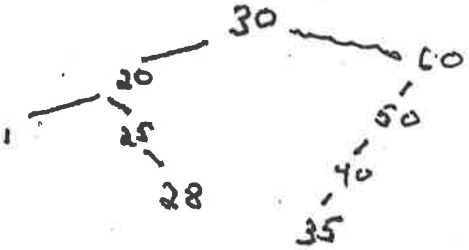
13, 40, 97, 7, 9, 2, 5, 8, 3

(d) In general, if a ternary (at most three children per node) tree is perfectly balanced (unlike the tree pictured here) and complete with height h, how many leaves, in terms of h, will the tree have? (2 points) 3^h Note, this tree has a perfectly flat bottom.

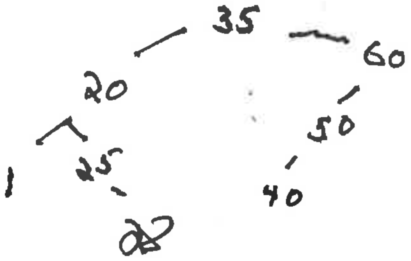
(e) In general, if a ternary tree is perfectly balanced (unlike the tree pictured here) and complete with exactly k leaves. What is the height (in terms of k) of this tree? (2 points) $Log_3 K$ Note, this tree has a perfectly flat bottom.

8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (3 Points)
30, 60, 20, 25, 28, 1, 50, 40, 35

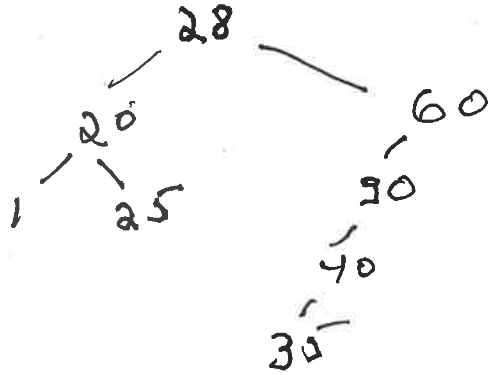




(b) Delete 30 from the final tree that you drew in 8 (a). Draw this final tree. (2 Points)
There are two possible answers. Either one is fine.



OR



-0

Key 5

Part 4. Project Questions (20 points)

(9) Recall the Merkle-Hellman cryptosystem that we worked with in Project 1.

Part of Project 1 was based on the subset sum problem which is known to be NP-Complete. The problem itself can be described as follows: given a set of numbers *X* and a number **k**, is there a subset of *X*, which sums to k?

(a) Suppose X = {56,3,9,12,4,2,234} and **k** = 15. Is there a subset of X which sums to k?
yes [✓ Yes]/No (1 points)

(b) The type of problem you were asked to solve in question 9 (a) is (Circle one answer): (1 Points)

1. an optimization problem.
2. a problem that is impossible to solve.
3. a problem that has been proven to take exponential time to solve.
4. a problem that has been proven to take factorial time to solve.
5. [✓ 5.] a decision problem.

(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (circle the one best option) (1 Points)

1. Find K so that M is prime.
2. Modify Bob's public key.
3. Modify the super increasing sequence.
4. Find a subset of a super increasing sequence that sums to K.
5. [✓ 5.] Find a subset of Bob's public key that sums to K.

(d) Recall that a modular inverse of an integer b mod m is the integer b^{-1} such that $(b^* b^{-1})$ mod m=1.
What is the modular inverse of 2 mod 13? __7__ (1 Points)

(e) Also, in Project 1, we built a Merkle Tree. If a program reads n transactions and builds a Merkle Tree from these n transactions, which one of the following statements is correct? Circle one statement. (3 Points)

1. [✓ 1.] The tree will be constructed in Θ(N) time.
2. The tree will be constructed in Θ(LogN) time
3. The tree will be constructed in Θ(N^2) time.
4. The tree will be constructed in Θ(1) time.
5. The tree will be constructed in Θ(N*LogN) time.

(f) In Project 2 we worked with crime records containing longitude and latitude values. The values were used for computing the distance between two crimes. Circle True or [✓ False]. (1 point)

(g) In Project 2 we worked with crime records containing state plane coordinates. The values were used for displaying markers in Google Earth. Circle True or [✓ False]. (1 point)

0

Key 6

(h) In Project 3 we wrote a Red Black binary search tree. Recall that we inserted variable names into the Red Black Tree. Let T(n) be the number of operations required to do the insert. In the worst case, which of the following are true about T(n)? Circle all of those that are true. (You may or may not have more than one answer.) (4 Points)

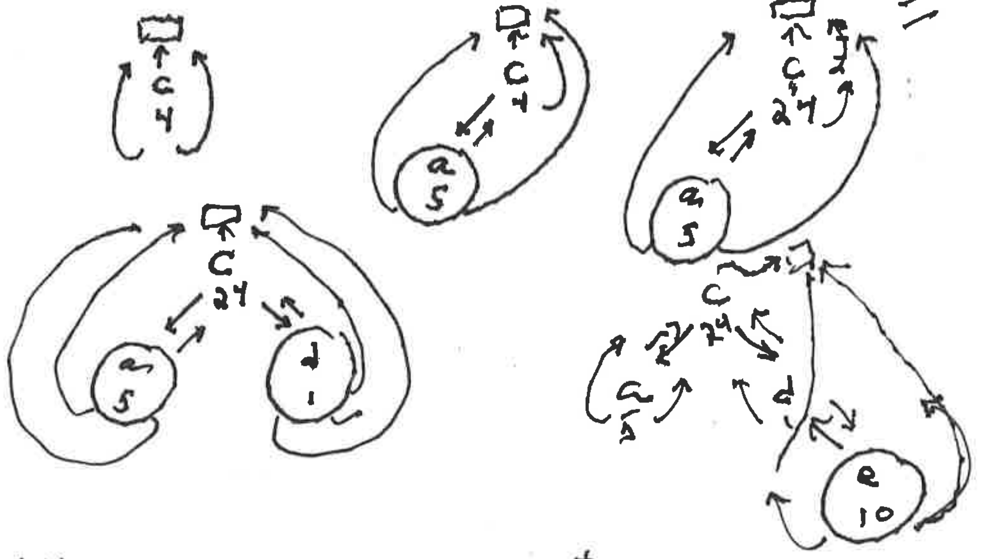
[✓ 1.] $T(n) \varepsilon O(LogN)$
2. $T(n) \varepsilon O(1)$
3. $T(n) \varepsilon \Omega(N^3)$
[✓ 4.] $T(n) \varepsilon O(N)$
[✓ 5.] $T(n) \varepsilon \Theta(LogN)$
[✓ 6.] $T(n) \varepsilon O(3^n)$
[✓ 7.] $T(n) \varepsilon \Omega(LogN)$
8. $T(n) \varepsilon \Theta(N)$

-1 for each error. Max off is 4.

Key 1

(1)The following is an execution of Project 3. Your task is to draw the Red Black Tree that exists after these commands are executed. Please be specific and show all of the content of each node. In addition, draw the nil node and show what nodes point to it. Color nodes red by drawing a circle around the node. (7 Points) 11

```
c 4 =
4
a 5 =
5
c a +
9
c c 20 + =
24
d 1 =
1
e 10 =
10
```

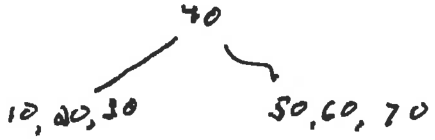


Part 5. B-Trees (21 points)

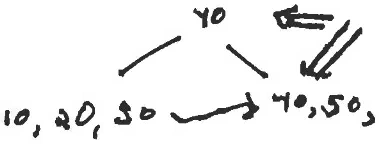
MAX = 6

10. a) Insert the following numbers into a B-Tree with a minimum of 3.
70,60,50,40,30,20,10. Draw the final tree. (7 Points)

10, 20, 30, 40, 50, 60, 70

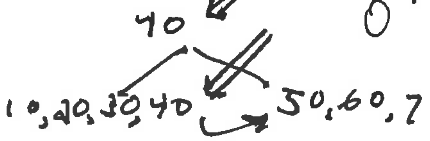


b) Insert the following numbers into a B+ Tree with a minimum of 3.
70,60,50,40,30,20,10. Draw the final tree. (7 Points)



OK

XOR



OK

c) Consider a B-Tree with a minimum of 2. What is the exact maximum number of keys such a tree could hold if the tree were of height 2? 124 (7 Points). Hint: A tree of height 0 has only a single root node with 4 keys.

0

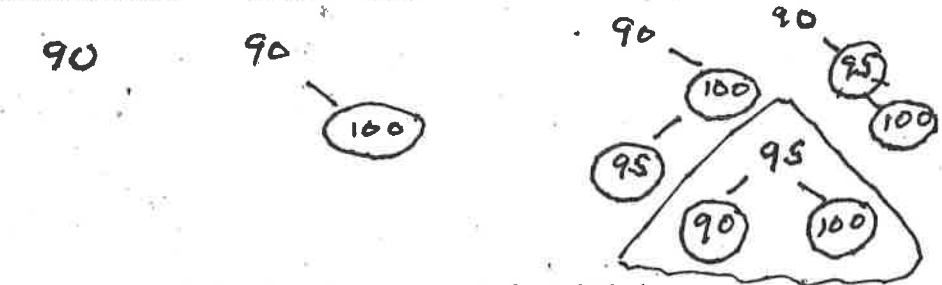
Key 8

Part 6. Red Black Trees (8 points)

11. Red-Black Trees

(a) Insert the following numbers, one by one, into a Red-Black Tree. Show the tree after each insertion. Draw RED nodes with a circle or a label 'R'. (5 points)

90, 100, 95



(b) When trying to analyze the run time complexity of an inorder traversal of a Red Black tree, one should consider the worst case, average case and ~~best case~~ separately. Each case may have a different run time performance. Circle TRUE or [✓ FALSE] (1 point)

(c) What is the worst-case runtime complexity of a Red Black Tree insert operation? Use Big Theta notation. (1 points) $\Theta(\log N)$

(d) What is the worst-case runtime complexity of a Binary Search Tree insert operation? Use Big Theta notation. (1 points) $\Theta(N)$

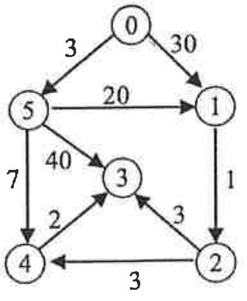
Part 7. Graph Algorithms (14 points)

See the attached graphs, G_1, G_2 and G_3.

12. (a) What is the shortest weighted path from the start node 0 to node 3 in the graph G_1? Your path must be a list of ordered pairs. (1 point) (0,1), (1,3)

(b) Show the list of nodes that would be visited by a breadth first search in the graph G_2. We are starting from vertex 0. (1 point) 0, {4, 2, 5}, {1, 3}, 6

(c) Draw the contents of the distance array for each iteration of Dijkstra's Algorithm as it works on the graph shown here on page 8 – not G_1, G_2 or G_3. The initial state is given. **Mark the node to be selected next to the left of the array (note how 0 is marked to the left of the first array.)** Fill in each array cell working downward. That is, complete the first column of arrays before the second column of arrays. (6 Points)



0	0	?	?	?	?	?
	0	1	2	3	4	5

0	30	?	?	?	3
0	1	2	3	4	5

0	23	?	43	10	3
0	1	2	3	4	5

0	23	?	12	10	3
0	1	2	3	4	5

0	23	?	12	10	3
0	1	2	3	4	5

0	23	24	12	10	3
0	1	2	3	4	5

(d) After Dijkstra is complete it also collects parent pointers for each node in the graph. Complete the chart below showing the parent of each node as computed by Dijkstra. (3 point)

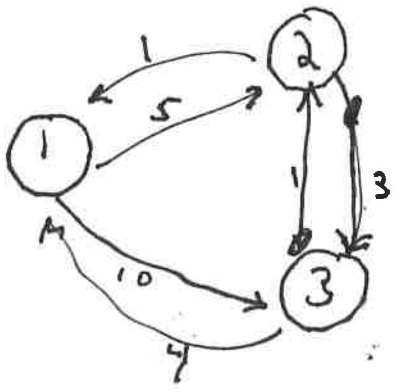
PARENT

Key 9

```
Node   Parent
===========
0       Nil
1       5
2       1
3       4
4       5
5       0
```

(i)Below is a recursive version of Floyd Warshall. Note: the real Floyd Warshall uses dynamic programming. Your problem is to show the output of this program. Note that it is too time consuming to trace the recursion. You will need to think about what Floyd Warshall does and compute the correct values yourself. The values -1 are used to signify that this column or row is unused. Java uses 0 indexing in arrays but Floyd Warshall describes the first node in the graph as node 1. The graph in this program corresponds to the graph G_3. Show the exact output as computed by this program. (3 Points)

```
public class FloydWarshallRecursive {


   public static int d(int w[][],int i, int j, int k){

      if ( k == 0) return w[i][j];
      else return Math.min(d(w,i,j,k-1),(d(w,i,k,k-1) + d(w,k,j,k-1)));
   }

   public static void main(String[] args) {


      int cost[][] = { {-1, -1, -1, -1}, {-1, 0, 5, 10}, {-1, 1, 0, 3}, {-1, 4 , 1, 0} };

      //display(cost);
      System.out.println(d(cost,1,2,3)); 5 // Display the cost from node 1 to node 2
      System.out.println(d(cost,1,3,3)); 8
      System.out.println(d(cost,2,1,3)); 1
      System.out.println(d(cost,2,3,3)); 3
      System.out.println(d(cost,3,1,3)); 2
      System.out.println(d(cost,3,2,3)); 1

   }

}
```



```
5
8
1
3
2
1
```

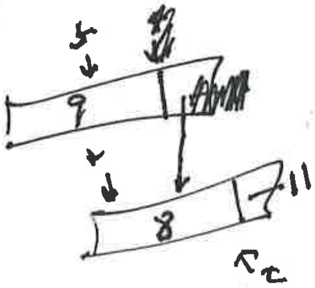
Key
10

Part 8. Queues (7 points)

13. Write a Queue class that is designed to process simple integers. Your Queue class will be based on a singly linked list. It will provide three methods – boolean emptyQueue(), void addAtRear(int x) and int removeFromFront(). The method emptyQueue() returns true if the queue has no data, false otherwise. These three methods must all run in constant time. Please use good syntax (not perfect but good), indentation and comments count for points.

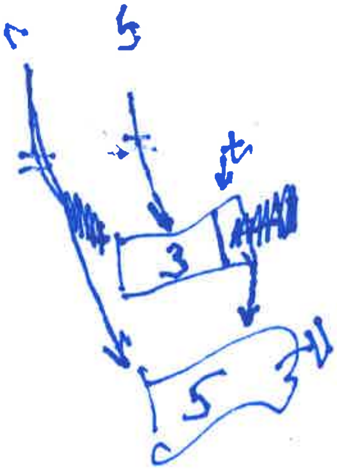
```
class Node {
    Node next
    int data
}
```

```
public class Queue {

    Node f, r;
    public Queue() {
        f = Null;
        r = Null;
    }
1   public boolean emptyQueue() {
        return f == Null;

    }
2   public void addAtRear(int x) {
        Node t = new Node(x)
        t.next = Null
        if (f == Null) {
            f = r = t;
        }
        else {
            r.next = t
            r = t;
        }
4   // pre: q is not empty
    int removeFromFront() {
        int t = f.data;
        f = f.next;
        ~~return t;~~
        if (f == Null) r = Null;
        return t;

}
```



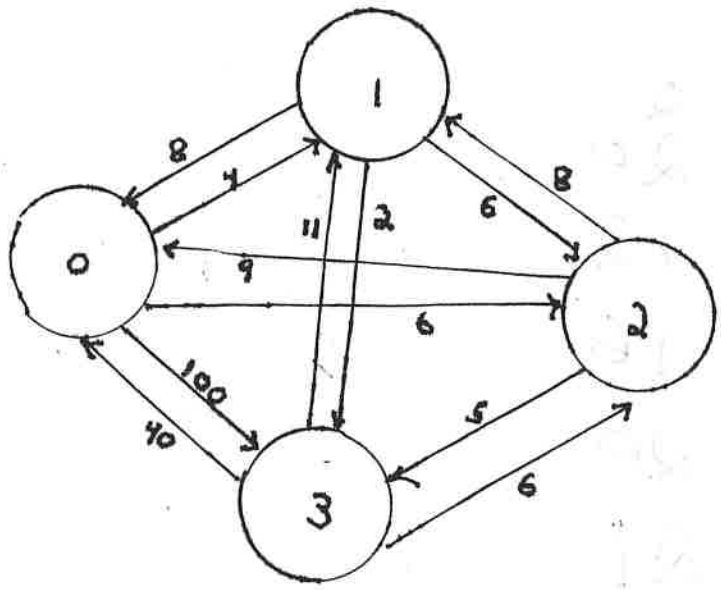
3

5

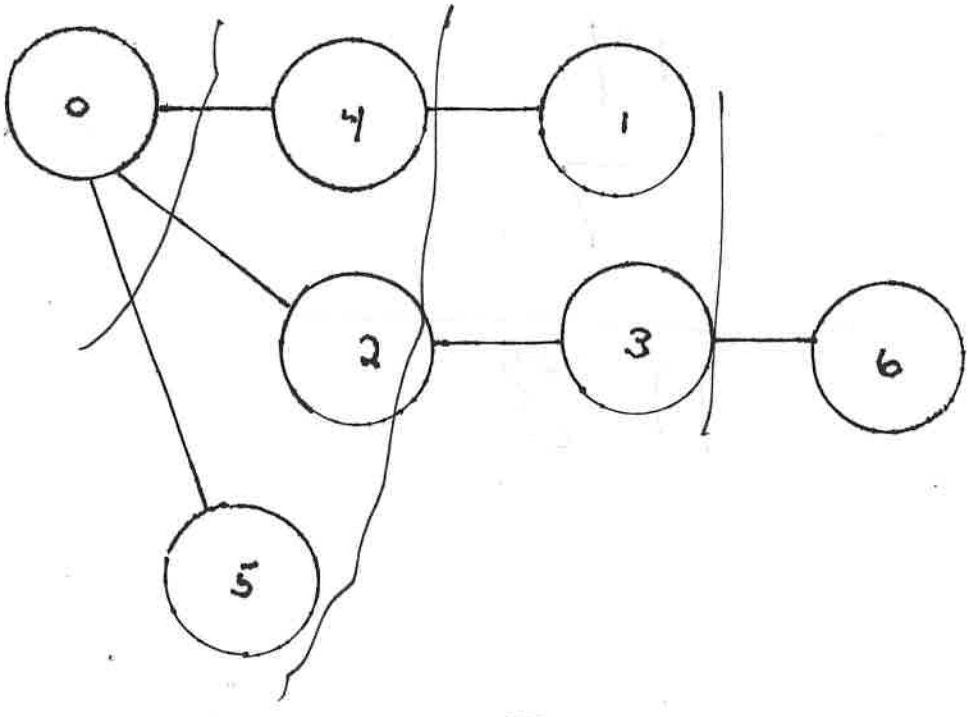


3

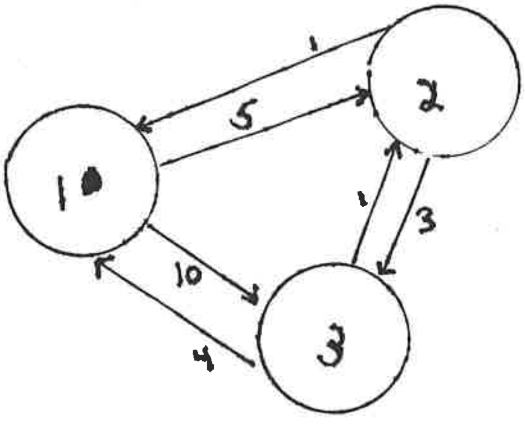
G_1



G_2



G_3


1
5
2
1
1
3
10
4
3