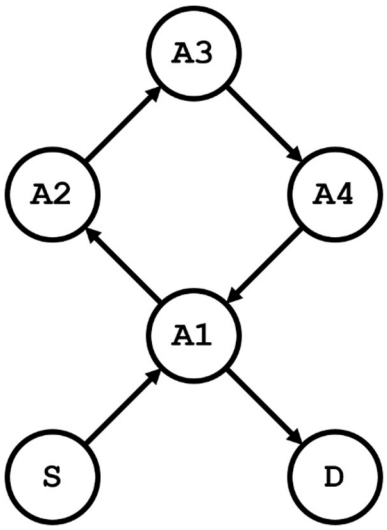
NAME: **Tyler Hu**

UH ID: **0276538**

COSC3320 – ASSIGNMENT 1

1. **(a) Background:**



The following algorithm implements a modified Towers of Hanoi game, represented by the graph G = (V, E) with V = {Start, Aux1, Aux2, Aux3, Aux4, Dest} and E = {(Start, Aux1), (Aux1, Aux2), (Aux2, Aux3), (Aux3, Aux4), (Aux4, Aux1), (Aux1, Dest)}.

The purpose of the game is to move n disks from the Start peg to the Dest peg via the Aux pegs. Only one disk can be moved at a time. Each move consists of taking the upper disk from one of the pegs and sliding it onto another pegs, on top of the other disks that may already be present on that tower. No disk can be placed on top of a smaller disk.

The algorithm can be implemented iteratively with the following pseudocode:

```
while true:
   if n is even:
      move disk1 one peg left (first peg wraps around to last peg)
   else:
      move disk1 one peg right (last peg wraps around to first peg)
   if done:
      break
   else:
      make the only legal move not involving disk1
```

Based on the stack size, the space complexity is $0(\mathrm{n})$ because the stack grows and shrinks linearly with n.

The total number of moves required is therefore $f(n) = 3\,f(n-1) + 2$. Solving this recursion formula will yield the solution $f(n) = 3^n - 1$. After this, it takes n moves to move all disks from the Start peg and another n moves to move all the disks to the Dest peg which yield us the time complexity of:

$$T(n) = 3^n - 1 + n + n = 3^n + 2n - 1$$

Therefore, the time complexity for this algorithm is $0(3^n)$.

Sources: https://www.math.toronto.edu/mathnet/questionCorner/genhanoi.html

https://stackoverflow.com/questions/12383044/complexity-for-towers-of-hanoi

(b) Methodology:

1. Assumptions:

   i. The disks are in order from smallest to largest and the pegs other than Start are empty.

   ii. All disks on the Start, Aux, and Dest pegs are correctly ordered during the game at all times.

2. The program was implemented in Java.

3. The program accepts an input consisting of an integer n, with a domain of $\{n \mid 1 \leq n \leq 10\}$, which is the number of disks used in the game.

4. First the Tower of Hanoi graph is built by a representation of nodes with corresponding “edges” simulated via next nodes.

5. The game is played for any amount of n disks by calling the load and unload functions to move disks from Start to Dest.

6. Note the Node class as a helper for the main Hanoi class.

Problem #1 in Java

Source: https://github.com/Ernaldis/Algorithms-3320/tree/master/Assignment1/Hanoi/src

```
import java.util.Stack;

public class Node {
    //this stack represents the contents of a peg in the game. Disks must be ints
    Stack<Integer> stack;
    Node next; //this will be used to more easily move nodes around the cycle
    String name; //this is used to denote the specific node

    /**
     * Initializer for Node
     * @param next the next node in the cycle
     * @param name the name of the node
     */
    Node(Node next, String name){
        this.next = next;
        this.name = name;
        this.stack = new Stack<>();
    }
}
```

```
package Hanoi;

public class hanoi {
    public static void main(String[] args) {
        //make the nodes of the graph
        Node start = new Node(null, "Start");
        Node node1 = new Node(null, "Aux1");
        Node node2 = new Node(null, "Aux2");
        Node node3 = new Node(null, "Aux3");
        Node node4 = new Node(null, "Aux4");
        Node dest = new Node(null, "Dest");
        //next nodes are used to move disks more easily
        start.next = node1;
        node1.next = node2;
        node2.next = node3;
        node3.next = node4;
        node4.next = node1;
        //placing the nodes in an array makes passing them as a parameter of the game
        //functions easier
        Node[] node = {start, node1, node2, node3, node4, dest};
        //play the first 10 games
        for (int i = 1; i <  2; i++){
            playGame(i, node);
            node[5].stack.clear();
        }
    }
    /**
     * Moves the disk on top of home's stack to the top of destination's stack
     * @param home the place the disk is before the function call
     * @param destination the place the disk is after the function call
     */
```

```
public static void move(Node home, Node destination){
    int disk = home.stack.pop();
    destination.stack.push(disk);
    System.out.println("Move disk #" + disk + " from " + home.name + " to " +
                       destination.name);
}
/**
 * Moves a stack of disks from one side of a cycle to the other. From node 2 to
 * node 4, for example.
 * @param disks: the number of disks in the stack to be moved
 * @param home: the node which the stack of disks is found
 */
public static void moveAcross(int disks, Node home){
    if (disks == 1){
        move(home,home.next);
        move(home.next,home.next.next);
    }
    else if (disks > 1) {
        moveAcross(disks - 1, home);
        move(home, home.next);
        moveAcross(disks - 1, home.next.next);
        move(home.next, home.next.next);
        moveAcross(disks - 1, home);
    }
}
/**
 * Plays a game for any amount of disks by calling the load and unload functions
 * @param disks the amount of disks in this game
 * @param node the nodes on which the game shall be played
 */
public static void playGame(int disks, Node[] node){
    System.out.println("Playing game for " + disks + " disks:");
    for (int i = 0; i < disks; i++){
        node[0].stack.push(disks-i);
    }
    load(node);
    unload(node);
}
/**
 * This function moves the stack from Start to Node 4.
 * @param node the set of nodes on which the game is being played.
 */
private static void load(Node[] node){
    while (!node[0].stack.isEmpty()){
        if(node[0].stack.size()==1){
            move(node[0],node[1]);
            move(node[1],node[5]);
        }
        else {
            move(node[0],node[1]);
            move(node[1],node[2]);
            move(node[2],node[3]);
            moveAcross(node[4].stack.size(), node[4]);
            move(node[3],node[4]);
            moveAcross(node[2].stack.size(), node[2]);
```

```
            }
        }
    }
    /**
     * This function moves the entire stack from Node 4 to Destination.
     * @param node The set of nodes on which the game was played.
     */
    private static void unload(Node[] node) {
        while (node[4].stack.size() != 0) {
            moveAcross(node[4].stack.size() - 1, node[4]);
            move(node[4], node[1]);
            move(node[1], node[5]);
            moveAcross(node[2].stack.size(), node[2]);
        }
        System.out.println("Completed game " + node[5].stack.size());
    }
}
```

Output:

```
Playing game for 1 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to destination
Completed game 1

Playing game for 2 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 2

Playing game for 3 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 3
```

```
Playing game for 4 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 4

Playing game for 5 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
```

```
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to destination
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 5


Playing game for 6 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
```

```
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to Node 2
Move disk #5 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
```

```
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
```

```
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 6
```

```
Playing game for 7 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to Node 2
Move disk #5 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
```

```
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
```

```
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 7
```

```
Playing game for 8 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
```

```
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
```

```
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to Node 2
Move disk #5 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
```

```
Move disk #1 from Node 1 to destination
```

```
Completed game 8
```

```
Playing game for 9 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to Node 2
Move disk #5 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
```

```
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 9

Playing game for 10 disks:
Move disk #1 from Start to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Start to Node 1
Move disk #2 from Node 1 to Node 2
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Start to Node 1
Move disk #3 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #4 from Start to Node 1
Move disk #4 from Node 1 to Node 2
Move disk #4 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
```

```
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #5 from Start to Node 1
Move disk #5 from Node 1 to Node 2
Move disk #5 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #4 from Node 4 to Node 1
Move disk #4 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #3 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 1 to Node 2
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #3 from Node 4 to Node 1
Move disk #3 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #2 from Node 2 to Node 3
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 3 to Node 4
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to Node 2
Move disk #2 from Node 4 to Node 1
Move disk #2 from Node 1 to destination
Move disk #1 from Node 2 to Node 3
Move disk #1 from Node 3 to Node 4
Move disk #1 from Node 4 to Node 1
Move disk #1 from Node 1 to destination
Completed game 10
```

2. **(a) in row-major order**

   For the 2-dimensional arrays A,B and C in row-major order, arrays A and B move from the first row to the last row, while from last to first for Array C. The code contains 3 reads (read from A, B, C) and 2 writes (write to A, B), for a total of 5 commands. Since each row and column fits in the memory (2000) in row-major order, the amount of transfers is:
   5 * 2000 = 10,000 transfers

   (b) in column-major order

   Every row and column has 2000 pages each. Again, the code contains 3 reads and 2 writes, for a total of 5 commands. Therefore the amount of transfers is:

   5 * 2000 * 2000 = 20,000,000 transfers

3. **(a)** If the pivot is always the first element in the list, the best scenario is when the pivot is the median of the list. This means that we want the first element on the list as the median of the whole list. Note that the domain is $\{n \mid 1 < n < \infty\}$.

   To have the array always be split equally by half, let the array be defined as: A[1:x] where as n approaches ∞, $x = n/2$. The best time complexity of QuickSort would thus be $\Omega(\mathrm{n}\ \log(\mathrm{n}))$.

   (b) QuickSort, despite the name, has three scenarios in which it runs with worst case time complexity: 1) where the array is already sorted, 2) where the array is sorted in reverse order, and 3) where all the elements in the array are the same.

   Since array A is to be sorted from smallest to largest element, all the smaller elements from the pivot x must be moved to the left of x in the array, while all the elements larger than x must be moved to the right of x.

   Again, note that the domain is $\{n \mid 1 < n < \infty\}$.

   However, since the pivot x is the first element, the worst-case time complexity occurs, since the pivot chosen is the first element of array A. The time complexity in this case would be 0(n^2) because the QuickSort algorithm would have to move every single element to the left of the pivot x for each recursive call of QuickSort.

   Source: https://www.geeksforgeeks.org/when-does-the-worst-case-of-quicksort-occur/

4. **Background:**

The workstation executing this program is an ASUS VivoBook K570UD with an Intel i5-8250U CPU, 8GB DDR4 RAM, GeForce GTX 1050 GPU, and Microsoft Windows 10 64-bit operating system.

Hypothesis:

The second allocation sequence of m arrays with 1million size will take longer than the first allocation sequence of $3m$ arrays with 800k size due to the defragmentation mechanism of memory trash collection.

Methodology:

1. The first step was to choose a proper m value. I started with $m = 1000$ to start with as a placeholder while I finished writing the program, but I was able to tweak the value to $m = 250{,}000$ because that value exhausts almost all of the memory allocated for the program without crashing. Note however, the program would still crash during some executions, which I believe is due to the fluctuations of computational resources allocated to other background operations in the workstation.
2. I initialized the first sequence of $3m$ arrays and used malloc to allocate 800k bytes to each element. I also initialized clock objects to calculate the time needed for allocation.
3. After initializing the arrays, I then deallocated memory to all even numbered arrays using the free operation.
4. Next I initialized the second sequence of m arrays and used malloc to allocate 1million bytes to each element. I also initialized clock objects to calculate the time needed for allocation.
5. I ran the program 10 times and got an average of 10.552 seconds for the 1st allocation sequence and 14.247 seconds for the 2nd allocation sequence.

Conclusion:

Based on the results of the experiment, I observed that the 2nd allocation took longer than the 1st allocation of arrays with $m = 250000$.

The results of the experiment support the hypothesis. As memory is released using the free() command in C, memory blocks are freed for use in the main memory. If a program attempts to provide memory blocks larger than those that have been left by the free() command, the computer attempts to provide as many as possible contiguous memory blocks that are right next to those particular positions in memory. Fragmentation occurs when programs are allocated memory blocks and but then they are not fully released after program execution ends. This is simulated by the allocation and deallocation of the even numbered arrays in the experiment. This leaves unused memory technically available, but if memory needs exceed the fragmented space's size, then that deallocated space is useless. This is solved by the defragmentation process (garbage collecting) of the memory by the OS where allocations are shifted to close gaps in memory to form contiguous memory blocks. The defragmentation process itself is what causes the 2nd allocation sequence of arrays to take significantly longer as shown in the result of the experiment.

Source: https://stackoverflow.com/questions/3770457/what-is-memory-fragmentation

Problem #4 in C

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MNUM 250000

//allocates arrays
void allocateArray(int **arr, int numOfArrays, int sizeOfArrays) {
    for (int i = 0; i < numOfArrays; i++) {
        arr[i] = (int*) malloc(sizeof(int) *sizeOfArrays);
    }
}

//loops through arrays and deallocates every even array
void deallocateEvenArrays(int **arr, int numOfArrays) {
    for (int i = 0; i < numOfArrays; i++) {
        if (i % 2 == 0) {
            free(arr[i]);
        }
    }
}

int main() {
    clock_t first_startT, first_endT;
    clock_t second_startT, second_endT;

    int **arr1;
    int **arr2;

    //first sequence starts
    first_startT = clock();

    //allocates 3*MNUM arrays
    arr1 = (int**) malloc(sizeof(int) * 3 * MNUM);
    allocateArray(arr1, 3 * MNUM, 800000);
    first_endT = clock();
    //first sequence ends

    deallocateEvenArrays(arr1, 3 * MNUM);

    //second sequence starts
    second_startT = clock();

    //allocates MNUM arrays
    arr2 = (int**) malloc(sizeof(int) * MNUM);
    allocateArray(arr2, MNUM, 1000000);
    second_endT = clock();
    //second sequence ends

    //prints out the time difference for both sequences & dealloc
    printf("FIRST ALLOCATION TIME: %f seconds\n", (double)(((double)first_endT -
       (double)first_startT)/(double)CLOCKS_PER_SEC));
     printf("SECOND ALLOCATION TIME: %f seconds\n", (double)(((double)second_endT -
       (double)second_startT)/(double)CLOCKS_PER_SEC));
    getchar();
    system(“pause”);

}
```

5. **Background:**

The following three programs implement a binary search algorithm for eight arrays of size 128, 512, 2048, 8192, 32768, 131072, 524288, and 2097152. The three distinct programming languages utilized were Python, Java, and C++.

Hypothesis:

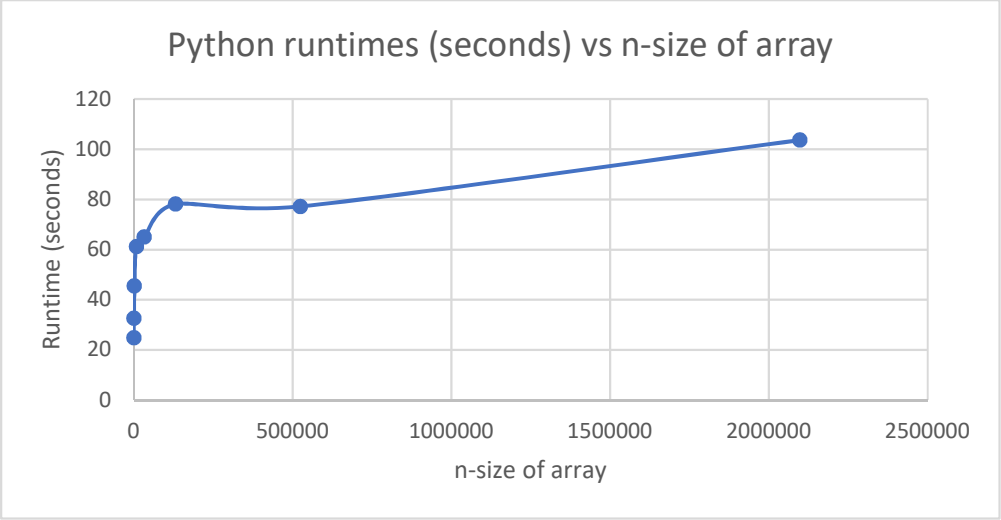
The time complexity of binary search is fastest in C++ and slowest in Python.

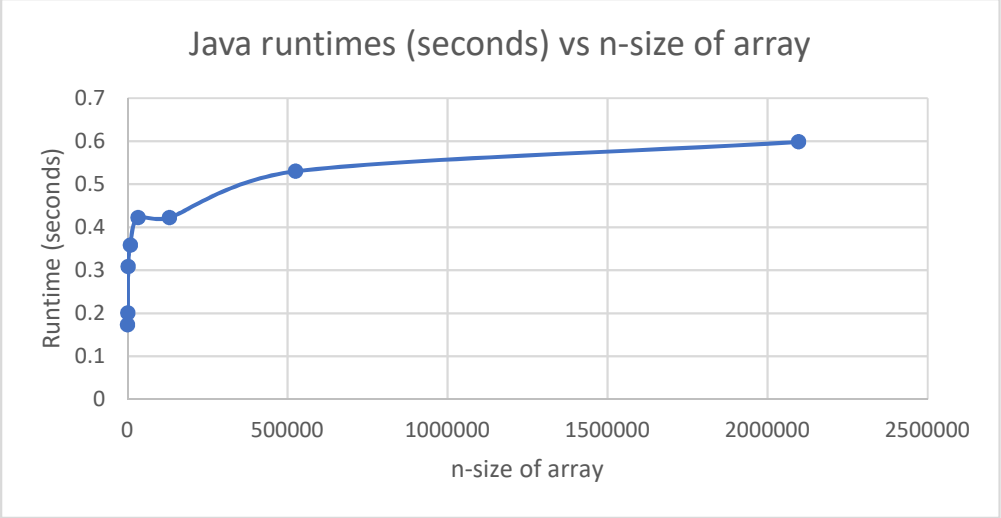
Methodology:

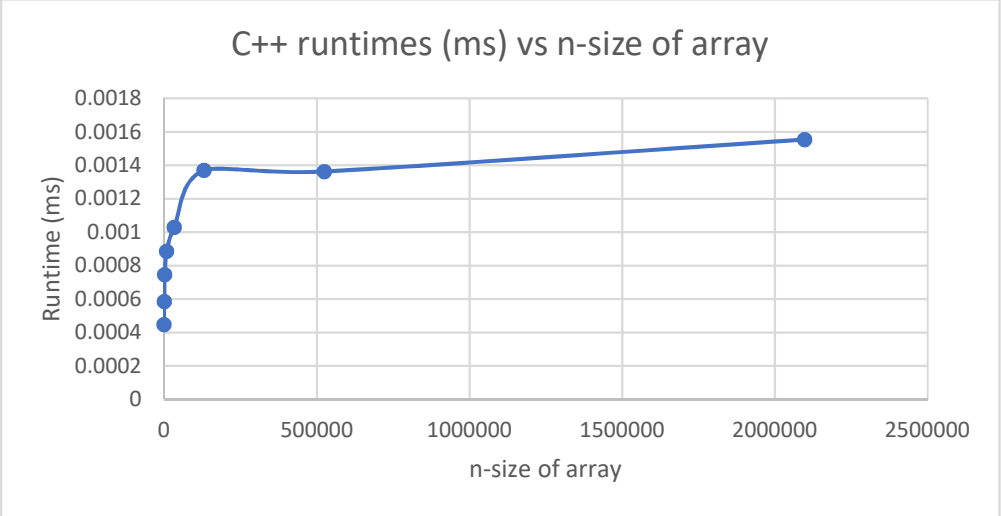
1. I first declared the eight arrays and filled them with integers from 0 to n (size of array).
2. Then the binary search function is passed a flag (2) so that it fails every time because the data is never 2.
3. I then made a loop of 10,000,000 unsuccessful searches and measured the time it took to execute it.
4. The runtimes for each program are shown in the table below.

n-size of Array	Python	Java	C++
128	24.83 seconds	0.17322 seconds	0.000000448 seconds
512	32.68 seconds	0.20086 seconds	0.000000585 seconds
2,048	45.56 seconds	0.30817 seconds	0.000000746 seconds
8,192	61.16 seconds	0.35829 seconds	0.000000885 seconds
32,728	64.99 seconds	0.42256 seconds	0.000001029 seconds
131,072	78.14 seconds	0.42256 seconds	0.000001369 seconds
524,288	77.21 seconds	0.52993 seconds	0.000001363 seconds
2,097,157	103.65 seconds	0.5983 seconds	0.000001554 seconds

5. I graphed the runtimes for each program as shown below.


Python runtimes (seconds) vs n-size of array
Runtime (seconds)
120
100
80
60
40
20
0
0
500000
1000000
1500000
2000000
2500000
n-size of array


Java runtimes (seconds) vs n-size of array
Runtime (seconds)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500000
1000000
1500000
2000000
2500000
n-size of array


C++ runtimes (ms) vs n-size of array
Runtime (ms)
0.0018
0.0016
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
0
500000
1000000
1500000
2000000
2500000
n-size of array

Conclusion:

Based on the table alone, we can only conclude that different programming languages yield different execution times for the same type of algorithm like binary search in this case. First let's compare the different execution times of binary search between C++, Java, and Python.

My hypothesis was indeed proven correct by the data. It is a well-known fact in the computing industry that C++ is one of the fastest languages on the market (besides FORTRAN), especially when compared to other languages such as Python. In the experiment, C++ was proven to have the quickest execution time because it is a language that is compiled during runtime and is more efficient at utilizing memory. Java was proven to be a bit slower than C++ because Java is also an interpreted language (has to be complied by a separate Java Virtual Machine instead of at runtime) with slower array access due to bound checks. Python was proven to have the slowest performance mainly because it is a high-level interpreted language ("translated" to binary at the time of execution instead of compiled at runtime). Python is considered a high-level language because of its dynamic features, such as run-time typing and hands-free memory allocation and garbage collection. However, since the language has to do more than say C++, it comes with the cost of increased overhead.

One thing of note is that the actual time complexity of binary search was proven to be $\theta(\log(n))$, regardless of programming language (see graphs). Even though python was significantly slower than Java and C++, all three languages demonstrated that the theoretical logarithmic time complexity of the binary search algorithm still held true.

Source: https://www.geeksforgeeks.org/binary-search/

Problem #5 in C++

```
#include<time.h>
#include<iostream>
using namespace std;

//binary search function that accepts an array, a value to find (flag), and
//values to find the midpoint (min and max)
int binarySearch(int arr[], int dataLocated, int min, int max) {
  while (max - min > 1) {
    int centerPos = (max - min)/2 + min;
    int center = arr[centerPos];

    if (dataLocated == center ) {
      return centerPos;
    }
    else if (dataLocated < center) {
      max = centerPos;
    }
    else if (dataLocated > center) {
      min = centerPos + 1;
    }
  }
  if (arr[min] == dataLocated)
    return min;
  else
    return -1;
}

//returns the array size given a base (2) and a power raised to
int power(int base, int powerRaised) {
  int result = 1;

  for (int i = 0; i < powerRaised; i++) {
    result = result = result*base;
  }
  return result;
}

int main() {
  int *myArrTestbed[8];
  //init arrays
  for (int i = 0; i < 8; i++) {
    myArrTestbed[i] = new int[power(2, 2*i+7)];
    for (int j = 0; j < power(2,2*i+7); j++) {
        myArrTestbed[i][j] = power (j,2);
    }
  }
  //searches arrays 10,000,000 times and prints benchmark times
  for (int i = 0; i < 8; i++) {
    clock_t start = clock();
    for (int j = 0; j < 10000000; j++) {
      binarySearch(myArrTestbed[i], 2, 0, power(2,(2*i+7)-1));
    }
    clock_t end = clock();
    cout << "The time needed to complete ten million iterations: " <<
            (end - start)*0.000000001 << " of size " << power(2,(2*i+7)) << endl;
  }
  for (int i = 0; i < 8; i++) {
    delete myArrTestbed[i];
  }
  return 0;
}
```

Problem #5 in Java

```
import java.lang.Math;

public class Main {
  //binary search function that accepts an array, a value to find (key), and
  //values to find the midpoint (min and max)
  static int binarySearch (int[] lst, int key, int min, int max) {
    while(max-min > 1) {
      int midPos = (max-min)/2 + min;
      int mid = lst[midPos];

      if (key == mid) {
        return midPos;
      }
      else if (key < mid) {
        max = midPos - 1;
      }
      else if (key > mid) {
        min = midPos + 1;
      }
    }

    if (lst[min] == key) {
      return min;
    }
    else {
      return -1;
    }
  }

  public static void main (String [] args) {
    //init arrays using power function from Math library
    int [][] myArrTestbed = new int [8][(int)Math.pow(2, (2*7+7))];
    for (int i = 0; i < 8; i++) {
      for(int j = 0; j < Math.pow(2, (2*i+7)); j++) {
        myArrTestbed[i][j] = (int)(Math.pow(j, 2));
      }
    }
    //searches arrays 10,000,000 times and prints benchmark times
    for(int i = 0; i < 8; i++) {
      long start = System.nanoTime();
      for (int j = 0; j < 10000000; j++) {
        binarySearch(myArrTestbed[i], 2, 0, (int) Math.pow(2, (2*i+7)-1));
      }
      long end = System.nanoTime();
      System.out.println((end-start)*Math.pow(10, -9) + " seconds for " +
                         (int)Math.pow(2,(2*i+7)));
    }
  }
}
```

Problem #5 in Python

```
import time

#binary search function that accepts an array, a value to find (key), and
#values to find the midpoint (min and max)
def binarySearch(lst, key, min, max):
  while max - min > 1:
    midPos = (max-min)//2+ min
    mid = lst[midPos]
    if key== mid:
        return midPos
    elif key < mid:
        max = midPos - 1
    elif key > mid:
        min = midPos + 1
  if lst[min]==key:
    return min
  else:
    return -1

#init arrays
myArrTestbed = [[],[],[],[],[],[],[],[]]
for i in range(8):
  for j in range(2**(2*i+7)):
    myArrTestbed[i].append(j**2)

#searches arrays 10,000,000 times and prints benchmark times
for i in range(8):
  start=time.time()
  for j in range(10000000):
    binarySearch(myArrTestbed[i], 2, 0, 2**(2*i+7)-1)
  end=time.time()
  size = len(myArrTestbed[i])
  print("The time for ten million iterations:", end - start, "seconds for", size)
```