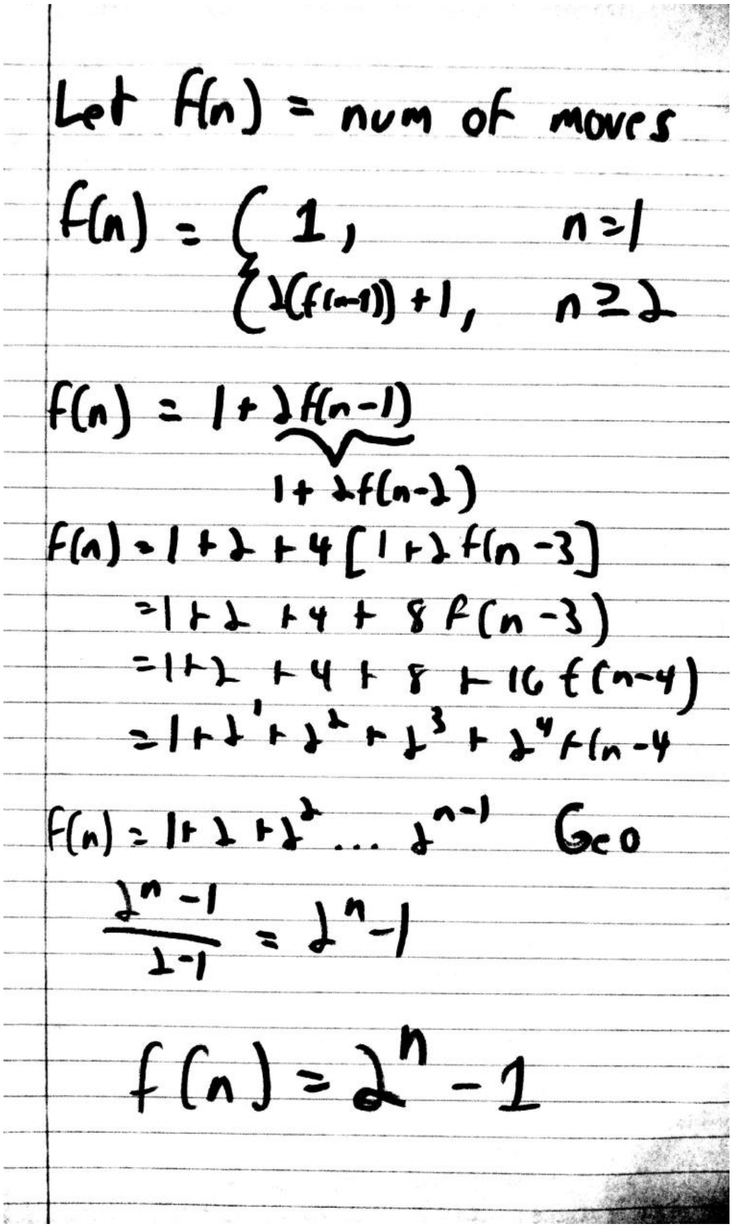
Brian Orozco March 31, 2017

Professor Nassimi CS – 241 – 002

Program 2: Towers of Hanoi

Values of f(n) for the values of n from 1 to 20:

N	F(n)
1	1
2	3
3	7
4	15
5	31
6	63
7	127
8	255
9	511
10	1023
11	2047
12	4095
13	8191
14	16383
15	32767
16	65535
17	131071
18	262143
19	524287
20	1048575



Let $f(n)$ = num of moves

$$f(n) = \begin{cases} 1, & n=1 \\ 2(f(n-1))+1, & n \geq 2 \end{cases}$$

$$f(n) = 1 + \underbrace{2f(n-1)}_{1+2f(n-2)}$$

$$f(n) = 1 + 2 + 4[1 + 2f(n-3)]$$

$$= 1 + 2 + 4 + 8f(n-3)$$

$$= 1 + 2 + 4 + 8 + 16f(n-4)$$

$$= 1 + 2^1 + 2^2 + 2^3 + 2^4 f(n-4$$

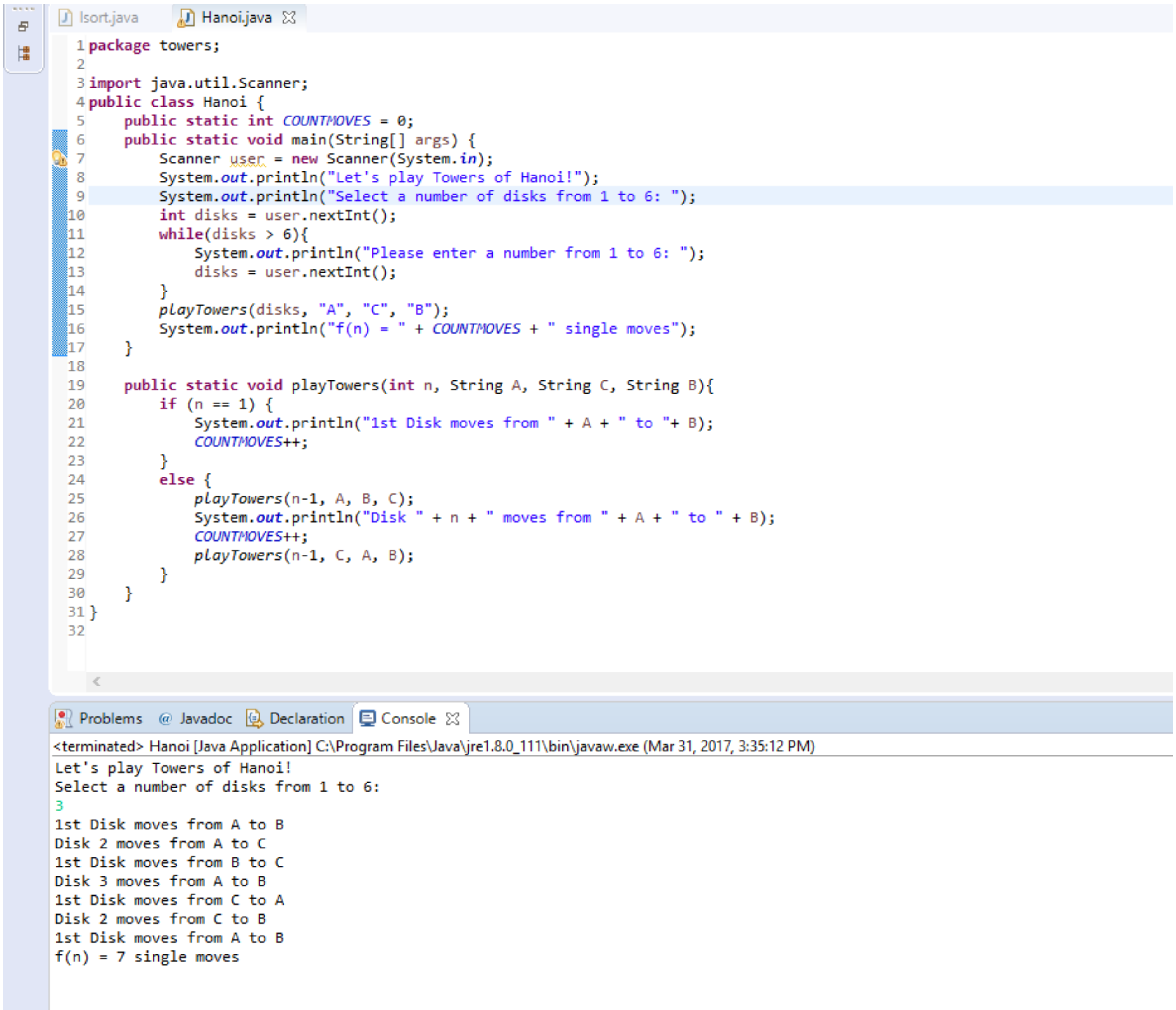
$$f(n) = 1 + 2 + 2^2 \ldots 2^{n-1}$$ Geo

$$\frac{2^n - 1}{2-1} = 2^n - 1$$

$$f(n) = 2^n - 1$$

Sample outputs:

- When n = 3

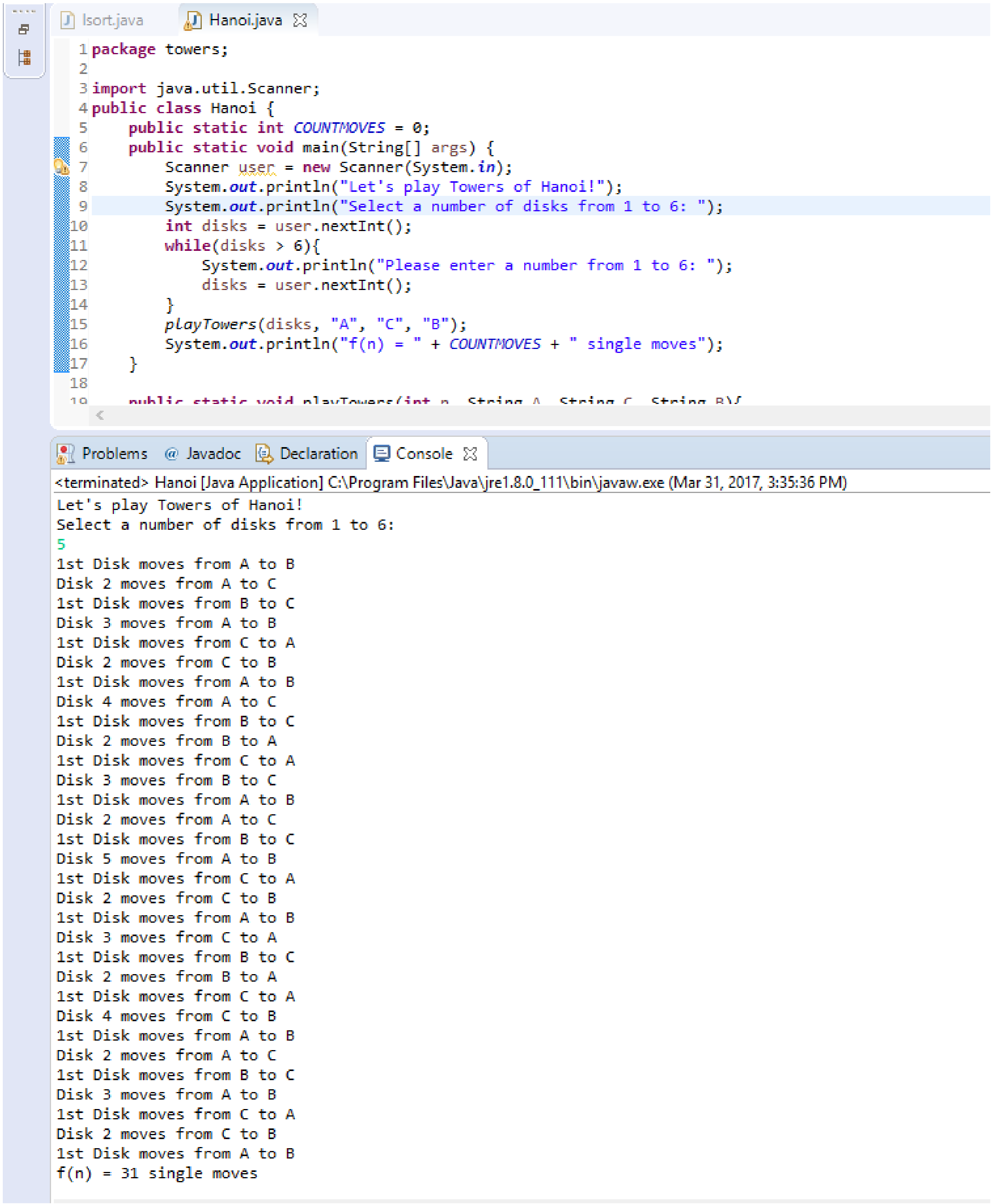


```java
package towers;

import java.util.Scanner;
public class Hanoi {
    public static int COUNTMOVES = 0;
    public static void main(String[] args) {
        Scanner user = new Scanner(System.in);
        System.out.println("Let's play Towers of Hanoi!");
        System.out.println("Select a number of disks from 1 to 6: ");
        int disks = user.nextInt();
        while(disks > 6){
            System.out.println("Please enter a number from 1 to 6: ");
            disks = user.nextInt();
        }
        playTowers(disks, "A", "C", "B");
        System.out.println("f(n) = " + COUNTMOVES + " single moves");
    }

    public static void playTowers(int n, String A, String C, String B){
        if (n == 1) {
            System.out.println("1st Disk moves from " + A + " to "+ B);
            COUNTMOVES++;
        }
        else {
            playTowers(n-1, A, B, C);
            System.out.println("Disk " + n + " moves from " + A + " to " + B);
            COUNTMOVES++;
            playTowers(n-1, C, A, B);
        }
    }
}

```

<terminated> Hanoi [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (Mar 31, 2017, 3:35:12 PM)

```
Let's play Towers of Hanoi!
Select a number of disks from 1 to 6: 
3
1st Disk moves from A to B
Disk 2 moves from A to C
1st Disk moves from B to C
Disk 3 moves from A to B
1st Disk moves from C to A
Disk 2 moves from C to B
1st Disk moves from A to B
f(n) = 7 single moves
```

- When n = 5



```java
package towers;

import java.util.Scanner;
public class Hanoi {
    public static int COUNTMOVES = 0;
    public static void main(String[] args) {
        Scanner user = new Scanner(System.in);
        System.out.println("Let's play Towers of Hanoi!");
        System.out.println("Select a number of disks from 1 to 6: ");
        int disks = user.nextInt();
        while(disks > 6){
            System.out.println("Please enter a number from 1 to 6: ");
            disks = user.nextInt();
        }
        playTowers(disks, "A", "C", "B");
        System.out.println("f(n) = " + COUNTMOVES + " single moves");
    }

    public static void playTowers(int n, String A, String C, String B){
```

```
<terminated> Hanoi [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (Mar 31, 2017, 3:35:36 PM)
Let's play Towers of Hanoi!
Select a number of disks from 1 to 6:
5
1st Disk moves from A to B
Disk 2 moves from A to C
1st Disk moves from B to C
Disk 3 moves from A to B
1st Disk moves from C to A
Disk 2 moves from C to B
1st Disk moves from A to B
Disk 4 moves from A to C
1st Disk moves from B to C
Disk 2 moves from B to A
1st Disk moves from C to A
Disk 3 moves from B to C
1st Disk moves from A to B
Disk 2 moves from A to C
1st Disk moves from B to C
Disk 5 moves from A to B
1st Disk moves from C to A
Disk 2 moves from C to B
1st Disk moves from A to B
Disk 3 moves from C to A
1st Disk moves from B to C
Disk 2 moves from B to A
1st Disk moves from C to A
Disk 4 moves from C to B
1st Disk moves from A to B
Disk 2 moves from A to C
1st Disk moves from B to C
Disk 3 moves from A to B
1st Disk moves from C to A
Disk 2 moves from C to B
1st Disk moves from A to B
f(n) = 31 single moves
```