Assignment 8
CS1083

Student Name: Omar Sebri
Student ID :3722350

Code:

```
public class InvalidSequenceTermException extends Exception{
    public InvalidSequenceTermException() {
        super("Invalid term");
    }
    public InvalidSequenceTermException(String message){
        super(message);
        }
}
```

```
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;
import java.lang.Math;
/**
A utility class that provide methods to compute elements of the
recursive sequence.
@author Omar Sebri
*/
public class Seq{

    /**
    Recursively computes seq(n).
    @param n Non-negative integer.
    @return int Element n in the recursive sequence.
    */
    private static ArrayList<Integer> cache = new
ArrayList<Integer>(Arrays.asList(1,5)) ;


    public static int seqR(int n){
        if (n==0)
            return 1;
        else if (n==1)
            return 5;
        else return(seqR(n-1)+seqR(n-2));
    }
    /* checks if the element is already in the arraylist if not
    elements up to n get calculated and added then seq(n) is returned */
    public static int seqM(int n){
        if (cache.size()+1>=n)
            return cache.get(n-1);
```

```
        else{
            int i=cache.size()-1;
            while(i<=n){
                cache.add(seqR(i+1));
                i++;
            }
            return cache.get(n);
        }
    }
    /* returns seq(n) by summing the last two elemnts in the array up until n
is reached
    the last element is then returned */
    public static int seqI(int n){
        int [] array = new int [n+1];
        array[0]=1;
        array[1]=5;
        int i=2;
        while(i<n+1){
            array[i]=array[i-1]+array[i-2];
            i++;
        }
        return array[array.length-1];
    }
    /* this method allows us to calculate the term n using the general term of
this
    recursive sequence */
    public static int seqMath(int n){
        double a;
        a=((0.5-(9/(2*Math.sqrt(5))))*Math.pow(((1-
Math.sqrt(5))/2),n))+((0.5+(9/(2*Math.sqrt(5))))*Math.pow(((1+Math.sqrt(5))/2)
,n));
        int b=(int)a;
        return b;
    }

}
```

```
import java.util.Scanner;
import java.text.NumberFormat;;

/**
A simple driver that uses the Seq class to compute the
nth element of the sequence.
@author Omar Sebri
*/

public class TestSeq{
```

```
    public static void main(String[] args){

        int n, seqRec, seqMem, seqIter, seqMa;

        Scanner scan = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");
        n = scan.nextInt();
        try{
        if(n<0)
            throw new InvalidSequenceTermException();

        seqRec = Seq.seqR(n);
        System.out.println("seqR(" + n + ") is: " + seqRec);


        seqMem = Seq.seqM(n);
        System.out.println("seqM(" + n + ") is: " + seqMem);


        seqIter = Seq.seqI(n);
        System.out.println("seqI(" + n + ") is: " + seqIter);

        seqMa = Seq.seqMath(n);
        System.out.println("seqMath(" + n + ") is: " + seqMa); }
        catch(InvalidSequenceTermException e){
            System.out.println(e.getMessage());
        }

        NumberFormat form = NumberFormat.getInstance();
        form.setMaximumFractionDigits(7);
        form.setMinimumFractionDigits(7);

        System.out.println("Execution Times in Milliseconds (ms)");
        System.out.println("Seq(n) \tRecursive \tMemoization \tItertive
\tMathematical");

        long start, end;
        int seqA;
        double time;
        for(int i = 20; i <= 40; i+=10){
        start = System.nanoTime();
        seqA = Seq.seqR(i);
        end = System.nanoTime();
        time = (double)(end-start)/1000000;
        System.out.print(i + "\t" + form.format(time));
        start = System.nanoTime();
        seqA = Seq.seqM(i);
        end = System.nanoTime();
        time = (double)(end-start)/1000000;
```

```
        System.out.print(i + "\t" + form.format(time));
        start = System.nanoTime();
        seqA = Seq.seqI(i);
        end = System.nanoTime();
        time = (double)(end-start)/1000000;
        System.out.print(i + "\t" + form.format(time));
        start = System.nanoTime();
        seqA = Seq.seqMath(i);
        end = System.nanoTime();
        time = (double)(end-start)/1000000;
        System.out.print(i + "\t" + form.format(time)+"\n");

        }
    }
}
```

Note: Being a math nerd, I was curious whether computing Seq(n) using its general term would be faster, so I calculated the general term, proved it through induction then made the method SeqMath(n) that returns seq(n).
Turns out for small elements there's no big difference but as soon as n gets bigger (in the order of the hundreds), computation time becomes even ten times faster than the iterative method !!!
Interesting isn't it ?

The proof can be found at the end of the report

Test Case 1:

Input : -1

Output:

Enter a positive integer: -1

Invalid term

Execution Times in Milliseconds (ms)

Seq(n)	Recursive	Memoization	Itertive	Mathematical
20	1.680700020	0,.560700020	0.006400020	0.,0794000
30	4.411300030	19.792300030	0.016600030	0.0040000
40	529.888700040	2,794.356500040	0.003600040	0.0022000

Test Case 2:

Input : 9

Output:

Enter a positive integer: 9

seqR(9) is: 191

seqM(9) is: 191

seqI(9) is: 191

seqMath(9) is: 191

Execution Times in Milliseconds (ms)

Seq(n) Recursive Memoization Itertive Mathematical

20 0,.163000020 0.224400020 0.003900020 0.0042000

30 5.033700030 22.172200030 0.002900030 0.0021000

40 526.329400040 2,418.911900040 0.003600040 0.0028000

Test Case 3:

I noticed that Memorization method is taking too long for the past 2 cases so I decided to run a test by inverting the for loop (i=40 then i=30 then i=20).
The Output Made sense.

The output:

Enter a positive integer: 9

seqR(9) is: 191

seqM(9) is: 191

seqI(9) is: 191

seqMath(9) is: 191

Execution Times in Milliseconds (ms)

Seq(n) Recursive Memoization Itertive Mathematical

40 533.567200040 2,135.600600040 0.004900040 0.0028000

30 5.097900030 0,019900030 0.010200030 0.0026000

20 0.043800020 0.005600020 0.002800020 0.0026000

we have :

$$u_0 = 1 , u_1 = 5$$

$$u_n = u_{n-1} + u_{n-2} \quad ; \forall n \geq 1$$

$$U_n = " A U_{n-1} + B u_{n-2}$$

$$A = 1 , B = 1$$

therefore this recursive sequence has the Caracteristic equation of $r^2 = Ar + B$

$$\Rightarrow r^2 = r + 1$$

$$\Rightarrow r^2 - r - 1 = 0$$

this equation have the roots : $r' = \frac{1-\sqrt{5}}{2}$ and $r'' = \frac{1+\sqrt{5}}{2}$

and so : $u_n = C.(r')^n + D.(r'')^n$

$$u_0 = 1 \Rightarrow C + D = 1$$

$$u_1 = 5 \Rightarrow C.\left(\frac{1-\sqrt{5}}{2}\right) + D\left(\frac{1+\sqrt{5}}{2}\right) = 5$$

$$\begin{cases} C + D = 1 \\ \frac{1}{2}C + \frac{1}{2}D - C\frac{\sqrt{5}}{2} + \frac{\sqrt{5}}{2}D = 5 \end{cases} \Rightarrow \begin{cases} C + D = 1 \\ \frac{1}{2} + \frac{\sqrt{5}}{2}(D - C) = 5 \end{cases}$$

$$\Rightarrow \begin{cases} C + D = 1 \\ \frac{\sqrt{5}}{2}(D - C) = \frac{9}{2} \end{cases} \Rightarrow \begin{cases} C + D = 1 \\ D - C = \frac{9}{\sqrt{5}} \end{cases}$$

$$\Rightarrow \begin{cases} C + D = 1 \\ D = \frac{1}{2} + \frac{9}{2\sqrt{5}} \\ C = \frac{1}{2} - \frac{9}{2\sqrt{5}} \end{cases}$$

therefore $u_n = \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n$

$30 + 24 = 54$
$6 \times 5 + 6 \times 4$
$6(5+4)$

Let's prove this through induction :

$$u_0 = \left(\frac{1}{2} - \cancel{\frac{9}{2\sqrt{5}}}\right) + \left(\frac{1}{2} + \cancel{\frac{9}{2\sqrt{5}}}\right)$$
$$= 1 : \text{true}$$

$$u_1 : \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right) + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)$$
$$= \cancel{\frac{1}{4}} - \cancel{\frac{\sqrt{5}}{4}} - \cancel{\frac{9}{4\sqrt{5}}} + \frac{9}{4} + \cancel{\frac{1}{4}} + \cancel{\frac{\sqrt{5}}{4}} + \cancel{\frac{9}{4\sqrt{5}}} + \frac{9}{4}$$
$$= \frac{1}{2} + \frac{9}{2} = 5 : \text{true}$$

Let's suppose $u_n = \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n$

Let's prove $u_{n+1} = \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n+1} + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}$

$$u_{n+1} = u_n + u_{n-1}$$
$$= \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n$$
$$+ \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n-1} + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}$$
$$= \cancel{\left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)\right)}$$
$$= \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\left(\frac{1-\sqrt{5}}{2} + 1\right) + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}\left(\frac{1+\sqrt{5}}{2} + 1\right)$$
$$= \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\left(\frac{6}{4} - \frac{2\sqrt{5}}{4}\right) + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}\left(\frac{6}{4} + \frac{2\sqrt{5}}{4}\right)$$
$$= \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\left(\frac{1-\sqrt{5}}{2}\right)^2 + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}\left(\frac{1+\sqrt{5}}{2}\right)^2$$
$$= \left(\frac{1}{2} - \frac{9}{2\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n+1} + \left(\frac{1}{2} + \frac{9}{2\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} : \text{Proved}$$