CS501
Mansi Shah
19526
Week3:Homework3

Q15 ==> Please compare the performance of iterative and recursive Fibonacci implementation

```java
package Week3;

import java.util.Scanner;

public class recursionVsIteration {

    private static long fibRecursion(int n) {
        if (n == 0)
            return 0;
        if (n == 1)
            return 1;
        return fibRecursion(n - 1) + fibRecursion(n - 2);
    }

    private static long fibIteration(int n) {
        long f[] = new long[n + 1];
        int i;
        f[0] = 0;
        f[1] = 1;

        for (i = 2; i <= n; i++) {
            f[i] = f[i - 1] + f[i - 2];
        }
        return f[n];
    }

    public static void main(String[] args) {
        long start, stop, result = 0;
        int n;
        Scanner reader = new Scanner(System.in);
        // nth element input
        System.out.print("Enter the last element of Fibonacci sequence: ");
        n = reader.nextInt();

        reader.close();

        // Print out iteration method
        System.out.println("Fibonacci iteration:");
        start = System.nanoTime();
        result = fibIteration(n);
        stop = System.nanoTime();
        System.out.println("Result: fibonacci of " + n + " is " + result + ".");
        System.out.println("Time to Calculate is :" + (stop - start) + "ns");

        // Print out recursive method
        System.out.println("Fibonacci recursion:");
        start = System.nanoTime();
        result = fibRecursion(n);
        stop = System.nanoTime();
        System.out.println("Result: Fibonacci of " + n + " is " + fibRecursion(n) + ".");
        System.out.println("Time to Calculate is :" + (stop - start) + "ns");
    }

}
```

Console Problems Debug Shell

<terminated> recursionVsIteration [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (Sep 24, 2019, 11:20:26 PM)

```
Enter the last element of Fibonacci sequence: 20
Fibonacci iteration:
Result: fibonacci of 20 is 6765.
Time to Calculate is :3900ns
Fibonacci recursion:
Result: Fibonacci of 20 is 6765.
Time to Calculate is :284800ns
```

```java
package Week3;

import java.util.Scanner;

public class recursionVsIteration {

    private static long fibRecursion(int n) {
        if (n == 0)
            return 0;
        if (n == 1)
            return 1;
        return fibRecursion(n - 1) + fibRecursion(n - 2);
    }

    private static long fibIteration(int n) {
        long f[] = new long[n + 1];
        int i;
        f[0] = 0;
        f[1] = 1;

        for (i = 2; i <= n; i++) {
            f[i] = f[i - 1] + f[i - 2];
        }
        return f[n];
    }

     public static void main(String[] args) {
         long start, stop, result = 0;
         int n;
         Scanner reader = new Scanner(System.in);
         // nth element input
         System.out.print("Enter the last element of Fibonacci sequence: ");
         n = reader.nextInt();

         reader.close();

         // Print out iteration method
         System.out.println("Fibonacci iteration:");
         start = System.nanoTime();
         result = fibIteration(n);
         stop = System.nanoTime();
         System.out.println("Result: fibonacci of " + n + " is " + result + ".");
         System.out.println("Time to Calculate is :" + (stop - start) + "ns");

         // Print out recursive method
         System.out.println("Fibonacci recursion:");
         start = System.nanoTime();
```

Console ⊠ Problems Debug Shell

<terminated> recursionVsIteration [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (Sep 24, 2019, 11:44:59 PM)

```
Enter the last element of Fibonacci sequence: 35
Fibonacci iteration:
Result: fibonacci of 35 is 9227465.
Time to Calculate is :5800ns
Fibonacci recursion:
Result: Fibonacci of 35 is 9227465.
Time to Calculate is :38021300ns
```

Element of Fibonacci Number:	20:(6765)	35:(9227465)
Iterative Time Calculate	3900 ns	5800 ns
Recursive Time Calculate	284800 ns	38021300 ns

Conclusion: When we compare Performance of Recursive Vs Iterative Recursive Method takes too much time compare with Iterative Method.