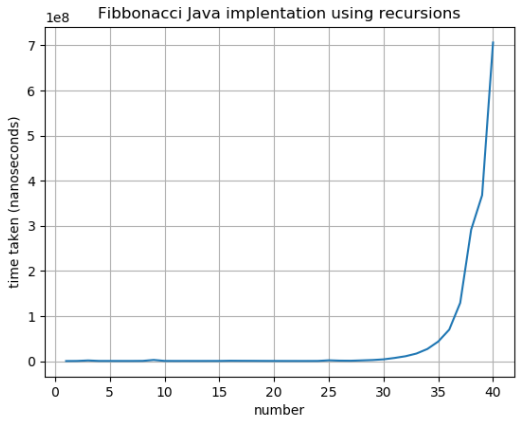
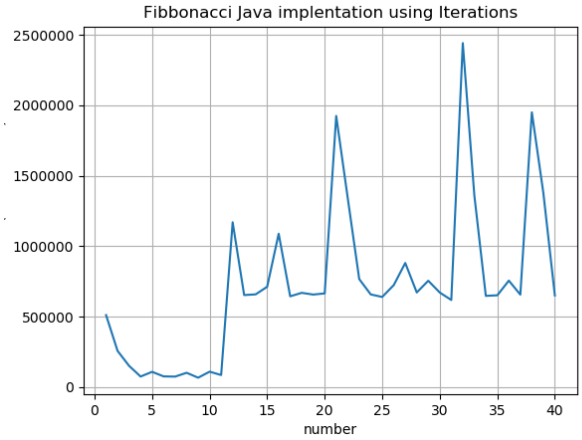
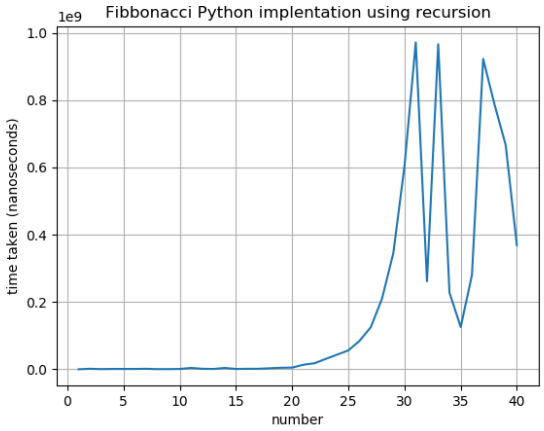
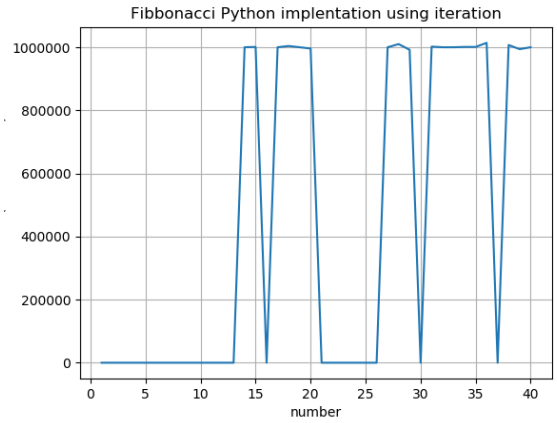








1. When looking at the outputs. Execution time of iteration and recursion in both Java and Python does not have a huge variation. Only iterative method using java have a variation other three methods have similar execution time when the problem is smaller.

   So it is clear that when the problem is small, which data structure or algorithm we use does not have a huge effect. But the when the problem is the data structure and the algorithm have a significant impact.

2. Both recursive method in both languages give same structure of output. Both iterative methods give fluctuating structure. Even though the values have some what difference the structure of output is same in both languages. So my idea is

Data structures and algorithms are concepts that are independent of language. Therefore once you master them in our favorite language it's relatively easy to switch to another one.

Now if you're asking about built-in methods and apis that different languages have, they do differ, but you shouldn't learn specific APIs in your data-structure and algorithms book anyways.

3. The difference in performance between Java and Python is sometimes significant in some cases.So there are differences between java and python.

   Performance is where Java has a substantial advantage over Python. Java's just-in-time compilation gives it an advantage over Python's interpreted performance. While neither language is suitable for latency-sensitive applications, Java is still a great deal faster than Python.

JVM (Java Virtual Machine) is an engine that gives runtime environment to operate the Java Code. It turns Java bytecode into machines language. JVM is a chunk of JRE (Java Run Environment). Python Interpreter translates your source code into machine-independent bytecode (. pyc).

4. When the problem is small iteration and recursion are both useful. When problem gets larger recursion might cause stack over flow. But recursive patter might be easy when implementing. Also time taken by the iteration is reduced when problem gets bigger. When problem is small both implementations are useful. But if both of these options are available it is better to use iterative algorithms.

5.Yes I agree. For the most part recursion is slower, and takes up more of the stack as well. The main advantage of recursion is that for problems like tree traversal it make the algorithm a little easier or more "elegant".

Try not to use recursion in system critical locations.
Elegant solutions not always the best performing when used in "recursive situations".
If you required to use recursion, at least try to optimize it with dynamic programming approaches (such as memorization). These facts should be considered when choosing a recursive algorithm.

JAVA CODE

```
import java.util.Arrays;

class Fib {

    public static int fib_r(int x) {
   if(x <= 2) return 1;
   return fib_r(x-1) + fib_r(x-2);
    }

    public static int fib_i(int x) {
   int a=1, b=1, fib=1, i=2;
   while(i<x) {
       fib = a + b;
       a = b;
       b = fib;
       i+=1;
   }
   return fib;
    }


    public static void main(String [] args) {
   int x = 10;
   int a[]=new int[40];
   float time[]=new float[40];


   /*for(int i=1;i<41;i++) {
      //measure time for executing using recursion java
      long startTime_r = System.nanoTime();
      System.out.println("Fib of " + i + " = " + fib_r(i));
      long stopTime_r = System.nanoTime();
      long elapsedTime_r = stopTime_r - startTime_r;
      a[i-1]=i;
      time[i-1]=elapsedTime_r;
      System.out.println("Java recursion time, number = " + i + " time in (nano sec) =
" + elapsedTime_r);

   }

      System.out.println(Arrays.toString(time));*/

      //measure time for executing using iteration java
      for(int j=1;j<41;j++) {
         long startTime_i = System.nanoTime();
         System.out.println("Fib of " + j + " = " + fib_i(j));
         long stopTime_i = System.nanoTime();
         long elapsedTime_i = stopTime_i - startTime_i;
         a[j-1]=j;
         time[j-1]=elapsedTime_i;
```

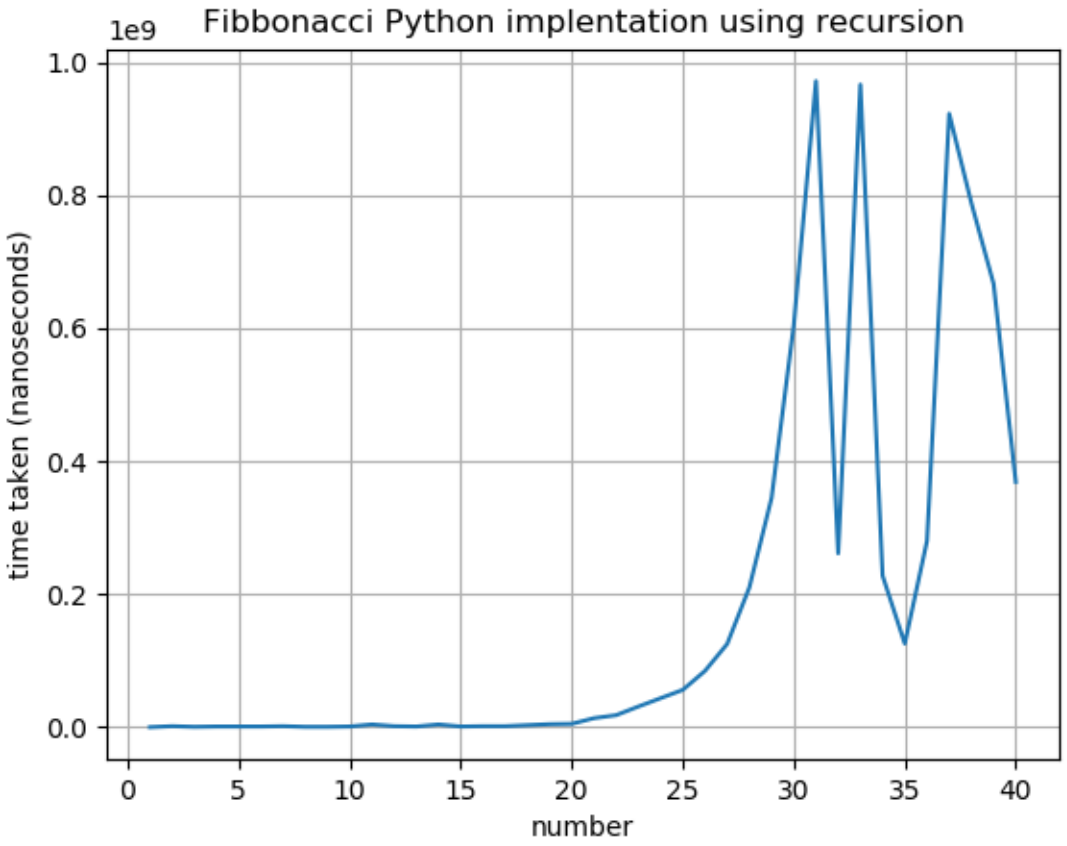
```
        System.out.println("Java iteration time, number = " + j + " time in (nano
sec) = " + elapsedTime_i);
      }

      System.out.println(Arrays.toString(time));
    }
}
```

PYTHON CODE

```
#!/usr/bin/python
import timeit
import datetime as dt
import matplotlib.pyplot as plt
import numpy as np


def fib_r(x):
    if (x <= 2):
        return 1
    return fib_r(x-1) + fib_r(x-2)


def fib_i(x):
    a = 1
    b = 1
    fib = 1
    i = 2
    while i < x:
        fib = a + b
        a = b
        b = fib
        i += 1

    return fib


"""
# measure time for recursion
a = []
time = []
for x in range(1, 41):
    start_r = dt.datetime.now()
```

```
    print("Fib of " + str(x) + " = " + str(fib_r(x)))
    end_r = dt.datetime.now()
    elapsed_r = (end_r - start_r).microseconds
    elapsed_r = elapsed_r*1000
    time.append(elapsed_r)
    print("Python recursion, number =" + str(x) + " time = "+str(elapsed_r))



print(*time)
list1 = range(1, 41)
print(*list1)

plt.plot(list1, time)
plt.xlabel('number')
plt.ylabel('time taken (nanoseconds)')
plt.title('Fibbonacci Python implentation using recursion')
plt.grid(True)
plt.show()
"""
a = []
time = []
# measure time for iteration
for x in range(1, 41):
    start_i = dt.datetime.now()
    print("Fib of " + str(x) + " = " + str(fib_i(x)))
    end_i = dt.datetime.now()
    elapsed_i = (end_i - start_i).microseconds
    elapsed_i = elapsed_i*1000
    time.append(elapsed_i)
    print("Python iterations, number =" + str(x) + " time = "+str(elapsed_i))

print(*time)
list1 = range(1, 41)
print(*list1)

plt.plot(list1, time)
plt.xlabel('number')
plt.ylabel('time taken (nanoseconds)')
plt.title('Fibbonacci Python implentation using iteration')
plt.grid(True)
plt.show()
```

RESULTS



Fib of 1 = 1

Python recursion, number =1 time = 0

Fib of 2 = 1

Python recursion, number =2 time = 1509000

Fib of 3 = 2

Python recursion, number =3 time = 499000

Fib of 4 = 3

Python recursion, number =4 time = 1000000

Fib of 5 = 5

Python recursion, number =5 time = 1001000

Fib of 6 = 8

Python recursion, number =6 time = 999000

Fib of 7 = 13

Python recursion, number =7 time = 1501000

Fib of 8 = 21

Python recursion, number =8 time = 494000

Fib of 9 = 34

Python recursion, number =9 time = 500000

Fib of 10 = 55

Python recursion, number =10 time = 1006000

Fib of 11 = 89

Python recursion, number =11 time = 4007000

Fib of 12 = 144

Python recursion, number =12 time = 1500000

Fib of 13 = 233

Python recursion, number =13 time = 1001000

Fib of 14 = 377

Python recursion, number =14 time = 4003000

Fib of 15 = 610

Python recursion, number =15 time = 1001000

Fib of 16 = 987

Python recursion, number =16 time = 1501000

Fib of 17 = 1597

Python recursion, number =17 time = 1498000

Fib of 18 = 2584

Python recursion, number =18 time = 3002000

Fib of 19 = 4181

Python recursion, number =19 time = 4497000

Fib of 20 = 6765

Python recursion, number =20 time = 5004000

Fib of 21 = 10946

Python recursion, number =21 time = 13511000

Fib of 22 = 17711

Python recursion, number =22 time = 18013000

Fib of 23 = 28657

Python recursion, number =23 time = 31014000

Fib of 24 = 46368

Python recursion, number =24 time = 43529000

Fib of 25 = 75025

Python recursion, number =25 time = 56040000

Fib of 26 = 121393

Python recursion, number =26 time = 84560000

Fib of 27 = 196418

Python recursion, number =27 time = 125089000

Fib of 28 = 317811

Python recursion, number =28 time = 210150000

Fib of 29 = 514229

Python recursion, number =29 time = 344742000

Fib of 30 = 832040

Python recursion, number =30 time = 604428000

Fib of 31 = 1346269

Python recursion, number =31 time = 971688000

Fib of 32 = 2178309

Python recursion, number =32 time = 261395000

Fib of 33 = 3524578

Python recursion, number =33 time = 966397000

Fib of 34 = 5702887

Python recursion, number =34 time = 227796000

Fib of 35 = 9227465

Python recursion, number =35 time = 125638000

Fib of 36 = 14930352

Python recursion, number =36 time = 279893000

Fib of 37 = 24157817

Python recursion, number =37 time = 922554000

Fib of 38 = 39088169

Python recursion, number =38 time = 788223000

Fib of 39 = 63245986

Python recursion, number =39 time = 666818000

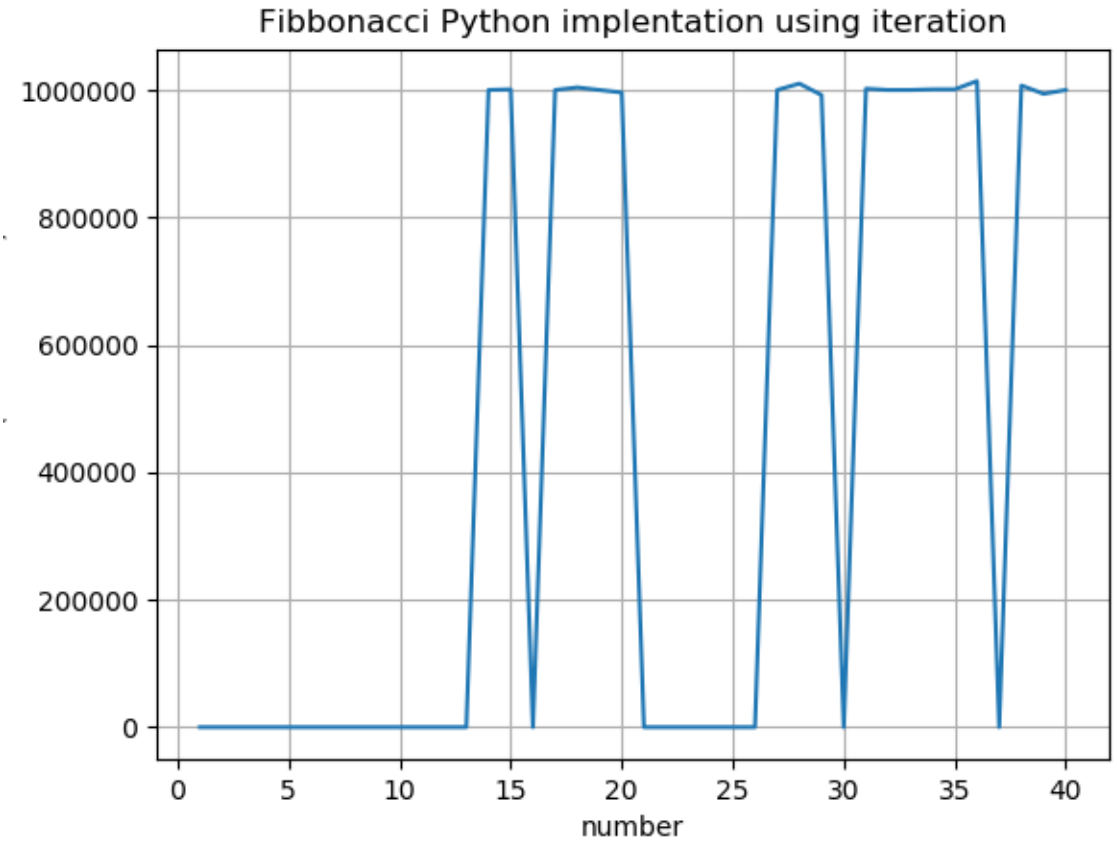
Fib of 40 = 102334155

Python recursion, number =40 time = 368818000

Two arrays

Number=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]

Time elapsed=[0 1509000 499000 1000000 1001000 999000 1501000 494000 500000 1006000 4007000 1500000 1001000 4003000 1001000 1501000 1498000 3002000 4497000 5004000 13511000 18013000 31014000 43529000 56040000 84560000 125089000 210150000 344742000 604428000 971688000 261395000 966397000 227796000 125638000 279893000 922554000 788223000 666818000 368818000]



Fib of 1 = 1

Python iterations, number =1 time = 0

Fib of 2 = 1

Python iterations, number =2 time = 0

Fib of 3 = 2

Python iterations, number =3 time = 0

Fib of 4 = 3

Python iterations, number =4 time = 0

Fib of 5 = 5

Python iterations, number =5 time = 0

Fib of 6 = 8

Python iterations, number =6 time = 0

Fib of 7 = 13

Python iterations, number =7 time = 0

Fib of 8 = 21

Python iterations, number =8 time = 0

Fib of 9 = 34

Python iterations, number =9 time = 0

Fib of 10 = 55

Python iterations, number =10 time = 0

Fib of 11 = 89

Python iterations, number =11 time = 0

Fib of 12 = 144

Python iterations, number =12 time = 0

Fib of 13 = 233

Python iterations, number =13 time = 0

Fib of 14 = 377

Python iterations, number =14 time = 1000000

Fib of 15 = 610

Python iterations, number =15 time = 1001000

Fib of 16 = 987

Python iterations, number =16 time = 0

Fib of 17 = 1597

Python iterations, number =17 time = 1000000

Fib of 18 = 2584

Python iterations, number =18 time = 1004000

Fib of 19 = 4181

Python iterations, number =19 time = 1000000

Fib of 20 = 6765

Python iterations, number =20 time = 996000

Fib of 21 = 10946

Python iterations, number =21 time = 0

Fib of 22 = 17711

Python iterations, number =22 time = 0

Fib of 23 = 28657

Python iterations, number =23 time = 0

Fib of 24 = 46368

Python iterations, number =24 time = 0

Fib of 25 = 75025

Python iterations, number =25 time = 0

Fib of 26 = 121393

Python iterations, number =26 time = 0

Fib of 27 = 196418

Python iterations, number =27 time = 1000000

Fib of 28 = 317811

Python iterations, number =28 time = 1010000

Fib of 29 = 514229

Python iterations, number =29 time = 992000

Fib of 30 = 832040

Python iterations, number =30 time = 0

Fib of 31 = 1346269

Python iterations, number =31 time = 1002000

Fib of 32 = 2178309

Python iterations, number =32 time = 1000000

Fib of 33 = 3524578

Python iterations, number =33 time = 1000000

Fib of 34 = 5702887

Python iterations, number =34 time = 1001000

Fib of 35 = 9227465

Python iterations, number =35 time = 1001000

Fib of 36 = 14930352

Python iterations, number =36 time = 1014000

Fib of 37 = 24157817

Python iterations, number =37 time = 0

Fib of 38 = 39088169

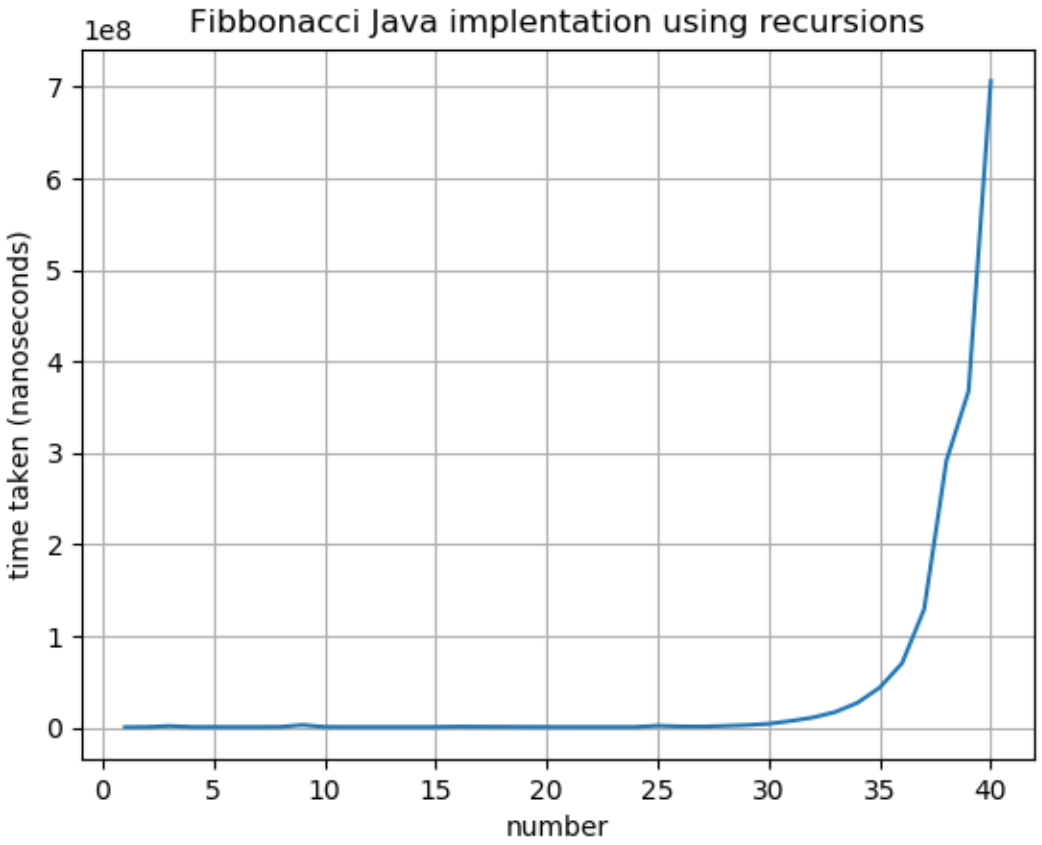
Python iterations, number =38 time = 1007000

Fib of 39 = 63245986

Python iterations, number =39 time = 994000

Fib of 40 = 102334155

Python iterations, number =40 time = 1000000



Fib of 1 = 1

Java recursion time, number = 1 time in (nano sec) = 461353

Fib of 2 = 1

Java recursion time, number = 2 time in (nano sec) = 632757

Fib of 3 = 2

Java recursion time, number = 3 time in (nano sec) = 1754065

Fib of 4 = 3

Java recursion time, number = 4 time in (nano sec) = 678431

Fib of 5 = 5

Java recursion time, number = 5 time in (nano sec) = 663034

Fib of 6 = 8

Java recursion time, number = 6 time in (nano sec) = 635836

Fib of 7 = 13

Java recursion time, number = 7 time in (nano sec) = 628651

Fib of 8 = 21

Java recursion time, number = 8 time in (nano sec) = 784146

Fib of 9 = 34

Java recursion time, number = 9 time in (nano sec) = 2854847

Fib of 10 = 55

Java recursion time, number = 10 time in (nano sec) = 723077

Fib of 11 = 89

Java recursion time, number = 11 time in (nano sec) = 638915

Fib of 12 = 144

Java recursion time, number = 12 time in (nano sec) = 663035

Fib of 13 = 233

Java recursion time, number = 13 time in (nano sec) = 619414

Fib of 14 = 377

Java recursion time, number = 14 time in (nano sec) = 659443

Fib of 15 = 610

Java recursion time, number = 15 time in (nano sec) = 671246

Fib of 16 = 987

Java recursion time, number = 16 time in (nano sec) = 1063318

Fib of 17 = 1597

Java recursion time, number = 17 time in (nano sec) = 935536

Fib of 18 = 2584

Java recursion time, number = 18 time in (nano sec) = 860611

Fib of 19 = 4181

Java recursion time, number = 19 time in (nano sec) = 743605

Fib of 20 = 6765

Java recursion time, number = 20 time in (nano sec) = 660469

Fib of 21 = 10946

Java recursion time, number = 21 time in (nano sec) = 604532

Fib of 22 = 17711

Java recursion time, number = 22 time in (nano sec) = 589136

Fib of 23 = 28657

Java recursion time, number = 23 time in (nano sec) = 581438

Fib of 24 = 46368

Java recursion time, number = 24 time in (nano sec) = 649692

Fib of 25 = 75025

Java recursion time, number = 25 time in (nano sec) = 2059924

Fib of 26 = 121393

Java recursion time, number = 26 time in (nano sec) = 1288607

Fib of 27 = 196418

Java recursion time, number = 27 time in (nano sec) = 1093083

Fib of 28 = 317811

Java recursion time, number = 28 time in (nano sec) = 1845412

Fib of 29 = 514229

Java recursion time, number = 29 time in (nano sec) = 2677285

Fib of 30 = 832040

Java recursion time, number = 30 time in (nano sec) = 4171679

Fib of 31 = 1346269

Java recursion time, number = 31 time in (nano sec) = 7244116

Fib of 32 = 2178309

Java recursion time, number = 32 time in (nano sec) = 11047842

Fib of 33 = 3524578

Java recursion time, number = 33 time in (nano sec) = 17015154

Fib of 34 = 5702887

Java recursion time, number = 34 time in (nano sec) = 27171079

Fib of 35 = 9227465

Java recursion time, number = 35 time in (nano sec) = 43756183

Fib of 36 = 14930352

Java recursion time, number = 36 time in (nano sec) = 70193922

Fib of 37 = 24157817

Java recursion time, number = 37 time in (nano sec) = 129347701

Fib of 38 = 39088169

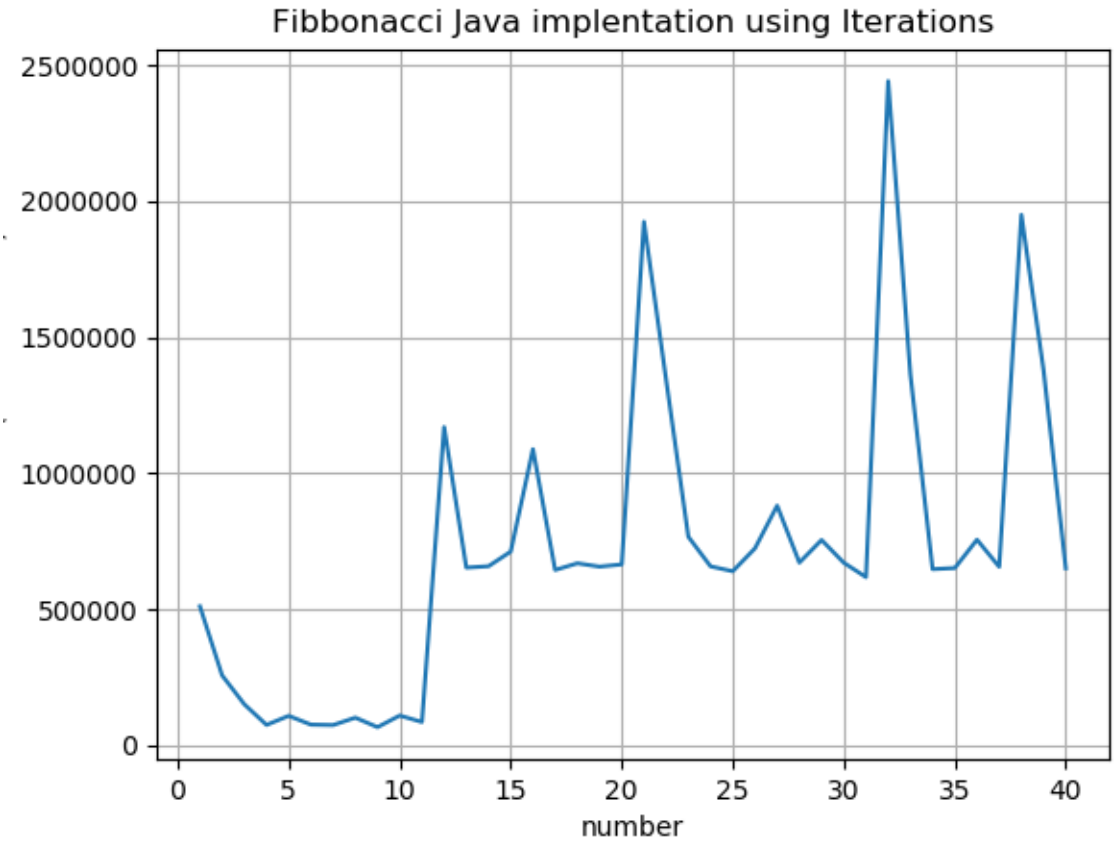
Java recursion time, number = 38 time in (nano sec) = 291462248

Fib of 39 = 63245986

Java recursion time, number = 39 time in (nano sec) = 367892905

Fib of 40 = 102334155

Java recursion time, number = 40 time in (nano sec) = 706380837



Fib of 1 = 1

Java iteration time, number = 1 time in (nano sec) = 481880

Fib of 2 = 1

Java iteration time, number = 2 time in (nano sec) = 795437

Fib of 3 = 2

Java iteration time, number = 3 time in (nano sec) = 777988

Fib of 4 = 3

Java iteration time, number = 4 time in (nano sec) = 744118

Fib of 5 = 5

Java iteration time, number = 5 time in (nano sec) = 727696

Fib of 6 = 8

Java iteration time, number = 6 time in (nano sec) = 650206

Fib of 7 = 13

Java iteration time, number = 7 time in (nano sec) = 683562

Fib of 8 = 21

Java iteration time, number = 8 time in (nano sec) = 641481

Fib of 9 = 34

Java iteration time, number = 9 time in (nano sec) = 867283

Fib of 10 = 55

Java iteration time, number = 10 time in (nano sec) = 1159797

Fib of 11 = 89

Java iteration time, number = 11 time in (nano sec) = 1793581

Fib of 12 = 144

Java iteration time, number = 12 time in (nano sec) = 1285015

Fib of 13 = 233

Java iteration time, number = 13 time in (nano sec) = 718972

Fib of 14 = 377

Java iteration time, number = 14 time in (nano sec) = 606584

Fib of 15 = 610

Java iteration time, number = 15 time in (nano sec) = 601966

Fib of 16 = 987

Java iteration time, number = 16 time in (nano sec) = 558345

Fib of 17 = 1597

Java iteration time, number = 17 time in (nano sec) = 618901

Fib of 18 = 2584

Java iteration time, number = 18 time in (nano sec) = 1910587

Fib of 19 = 4181

Java iteration time, number = 19 time in (nano sec) = 653284

Fib of 20 = 6765

Java iteration time, number = 20 time in (nano sec) = 1511842

Fib of 21 = 10946

Java iteration time, number = 21 time in (nano sec) = 614796

Fib of 22 = 17711

Java iteration time, number = 22 time in (nano sec) = 621467

Fib of 23 = 28657

Java iteration time, number = 23 time in (nano sec) = 1229078

Fib of 24 = 46368

Java iteration time, number = 24 time in (nano sec) = 673812

Fib of 25 = 75025

Java iteration time, number = 25 time in (nano sec) = 647639

Fib of 26 = 121393

Java iteration time, number = 26 time in (nano sec) = 664575

Fib of 27 = 196418

Java iteration time, number = 27 time in (nano sec) = 1951642

Fib of 28 = 317811

Java iteration time, number = 28 time in (nano sec) = 1222919

Fib of 29 = 514229

Java iteration time, number = 29 time in (nano sec) = 650718

Fib of 30 = 832040

Java iteration time, number = 30 time in (nano sec) = 686641

Fib of 31 = 1346269

Java iteration time, number = 31 time in (nano sec) = 677917

Fib of 32 = 2178309

Java iteration time, number = 32 time in (nano sec) = 713840

Fib of 33 = 3524578

Java iteration time, number = 33 time in (nano sec) = 952984

Fib of 34 = 5702887

Java iteration time, number = 34 time in (nano sec) = 714866

Fib of 35 = 9227465

Java iteration time, number = 35 time in (nano sec) = 1225485

Fib of 36 = 14930352

Java iteration time, number = 36 time in (nano sec) = 686641

Fib of 37 = 24157817

Java iteration time, number = 37 time in (nano sec) = 719485

Fib of 38 = 39088169

Java iteration time, number = 38 time in (nano sec) = 2269303

Fib of 39 = 63245986

Java iteration time, number = 39 time in (nano sec) = 672272

Fib of 40 = 102334155

Java iteration time, number = 40 time in (nano sec) = 670732