

## Math 188, Winter 2001, Prof. Tesler

### 3 algorithms to compute Fibonacci numbers, in C and in Java, run on 2 computers

```
/* Math 188, Winter 2001, Prof. Tesler */
/* Fibonacci numbers, computed by 3 algorithms written in C */

#include <stdio.h>
#include <math.h>

typedef long FN;

/* Fibonacci algorithm 1:
   Recurse w/o saving intermediate results. Exponential time. */
FN Fibonacci1(int n) {
    if (n<2)
        return 1;
    else
        return Fibonacci1(n-1)+Fibonacci1(n-2);
}

/* Fibonacci algorithm 2:
   Iteration. Linear time. */
FN Fibonacci2(int n) {
    FN a=1, b=1, c=1;    /* a,b are preceeding terms,
                        c is next term */

    int i;

    for(i=2; i<=n; i++) {
        c = a+b;        /* compute next term */
        a=b;            /* shift terms by 1 */
        b=c;
    }
    return c;
}

/* Fibonacci algorithm 3:
   Direct formula using golden ratio. Constant time. */
FN Fibonacci3(int n) {
    double sq5 = sqrt((double)5);
    double gr1 = (1+sq5)/2;    /* golden ratio */
    double gr2 = (1-sq5)/2;

    /* The formula  $Fib(n) = (gr1^{n+1} - gr2^{n+1}) / \sqrt{5}$ 
       is exact, but floating point arithmetic is not, so round. */
    return (FN) floor((pow(gr1,n+1) - pow(gr2,n+1))/sq5 + 0.5);
}

/* Print table of first 50 Fibonacci numbers, using one of above algorithms */
void PrintFibs(FN (*Fib)(int), char *Fname) {
    int n;
    printf("n\t%s(n)\n", Fname);

    for(n=0; n<50; n++)
        printf("%d\t%d\n", n, (*Fib)(n));
}

int main(int argc, char *argv[]) {
    PrintFibs(&Fibonacci1, "Fib1");
}
```

```
// Math 188, Winter 2001, Prof. Tesler
// Fibonacci numbers, computed by 3 algorithms written in Java

// An interface defining the format of all the Fib functions:
interface Fibo {
    public long Fib(int n);
}

// Fibonacci algorithm 1:
// Recurse w/o saving intermediate results. Exponential time.
class Fibonacci1 implements Fibo {
    public long Fib(int n) {
        return (n<2) ? 1 : Fib(n-1)+Fib(n-2);
    }
}

// Fibonacci algorithm 2:
// Iteration. Linear time.
class Fibonacci2 implements Fibo {
    public long Fib(int n) {
        long a = 1, b = 1; // last two Fib #'s
        long c;           // and the next one

        for (int i = 2; i <= n; i++) {
            c = a+b;      // compute next term
            a = b;        // shift two term window
            b = c;
        }
        return b;
    }
}

// Fibonacci algorithm 3:
// Direct formula using golden ratio. Constant time.
class Fibonacci3 implements Fibo {
    public long Fib(int n) {
        double Sq5 = Math.sqrt(5);
        double GR1 = (1+Sq5)/2;
        double GR2 = (1-Sq5)/2;

        return Math.round((Math.pow(GR1,n+1)
            - Math.pow(GR2,n+1))/Sq5);
    }
}

// Print table of first 50 Fibonacci numbers, using one of above algorithms
class PrintFibs {
    public static void Print50(Fibo cl, String Fname) {
        System.out.println("n\t" + Fname + " (n)");

        for(int n=0; n<50; n++)
            System.out.println("  " + n + "\t" + cl.Fib(n));
    }

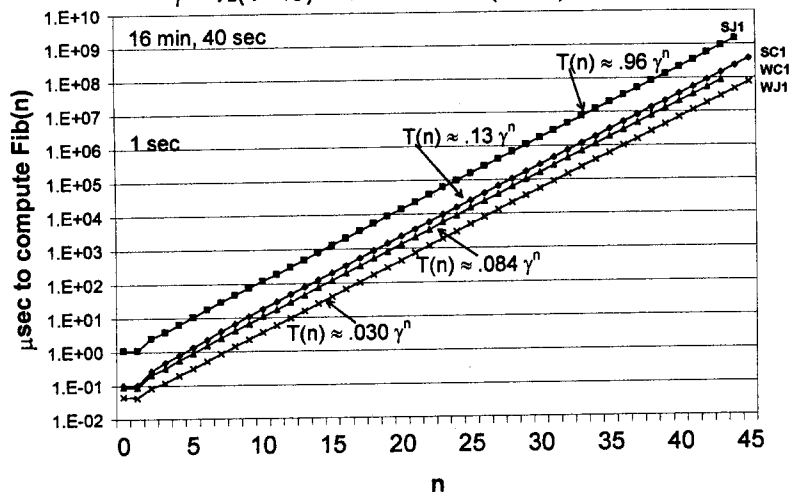
    public static void main(String[] args) {
        PrintFibs.Print50(new Fibonacci1(), "Fib1");
    }
}
```

## Timings by algorithm

### Fibonacci1 Timings

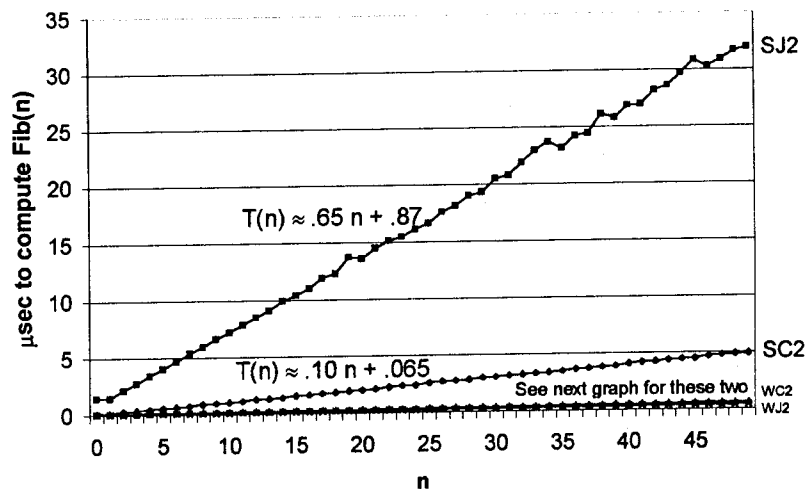
$$T(n) = K_1 \gamma^n + K_2 \delta^n + K_3 \in \Theta(\gamma^n)$$

$$\gamma = \frac{1}{2}(1+\sqrt{5}) \approx 1.62 \quad \delta = \frac{1}{2}(1-\sqrt{5}) \approx -.62$$



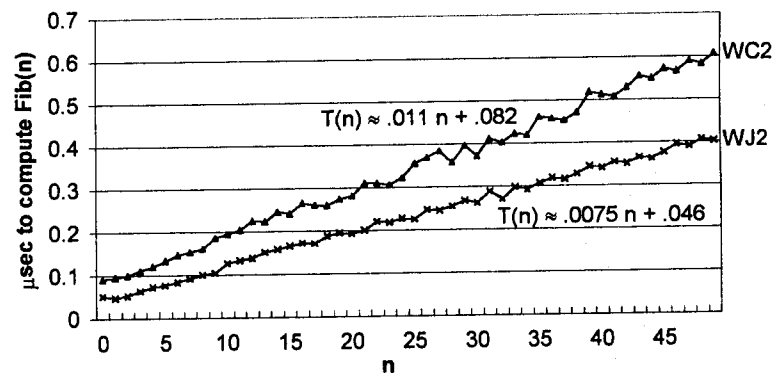
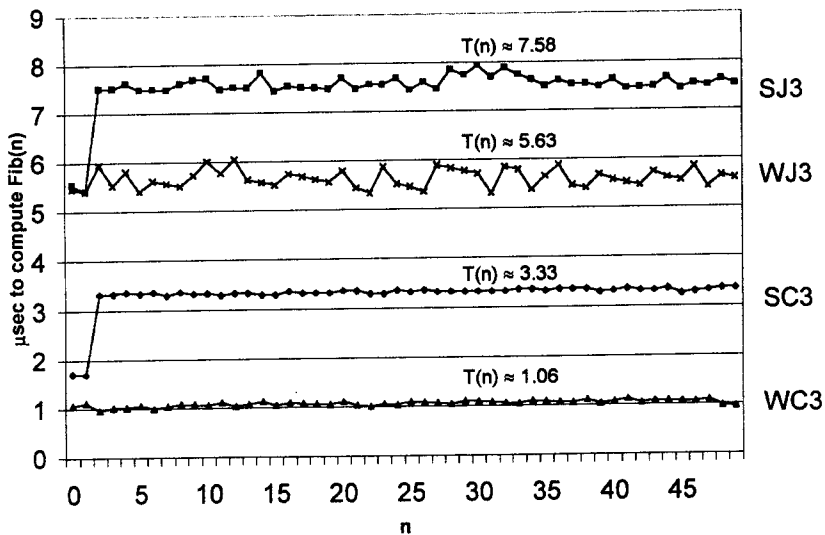
### Fibonacci2 Timings

$$T(n) = K_1 n + K_2 \in \Theta(n)$$



### Fibonacci3 Timings

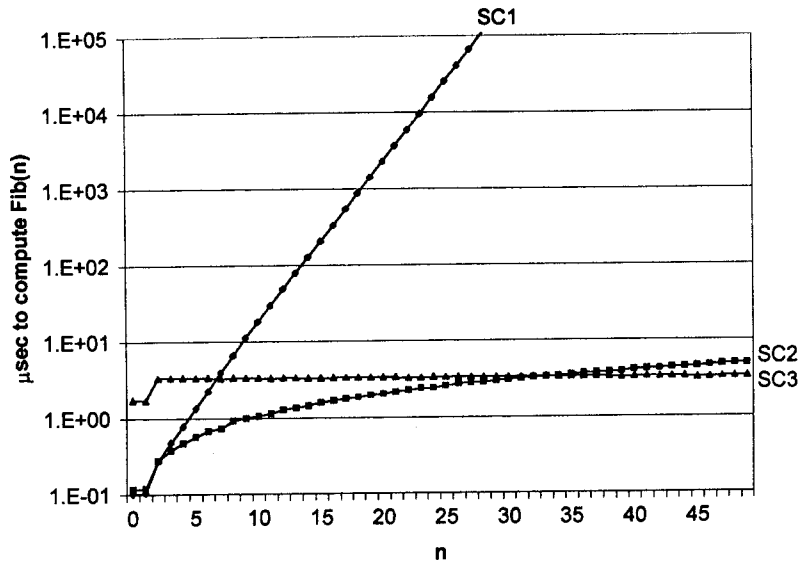
$$T(n) = K \in \Theta(1)$$



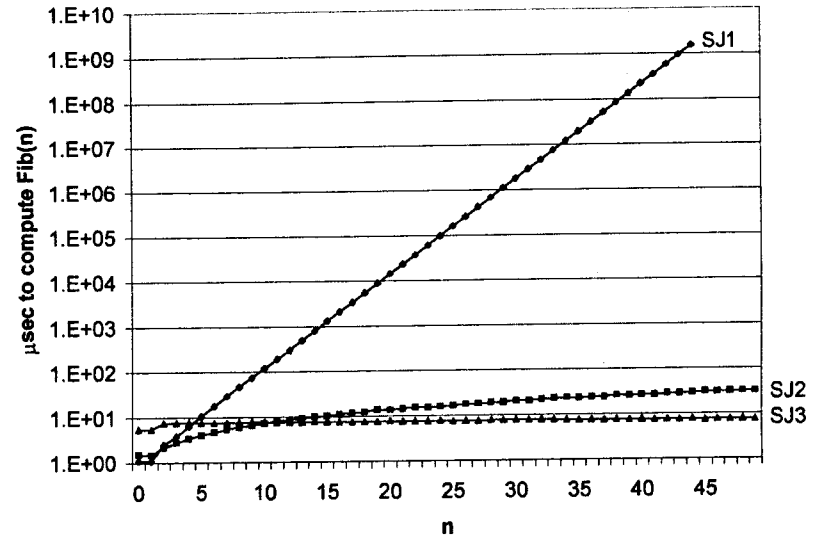
**Key:** All 12 combinations of 2 platforms, 2 languages, 3 algorithms  
**Platforms:** W = Windows ME, Athlon K7, 908 MHz; MS Visual Studio 6.0  
 S = Sun UltraSPARC2, Unix, 296 MHz; Gnu C 2.7, Sun JDK 1.1.6  
**Languages:** C, J=Java **Algorithms:** Fibonacci 1,2,3  
**Naming the 12 combinations of these:**  
 WJ2 = timings for Windows running Java implementation of Fibonacci2

# Timings by platform

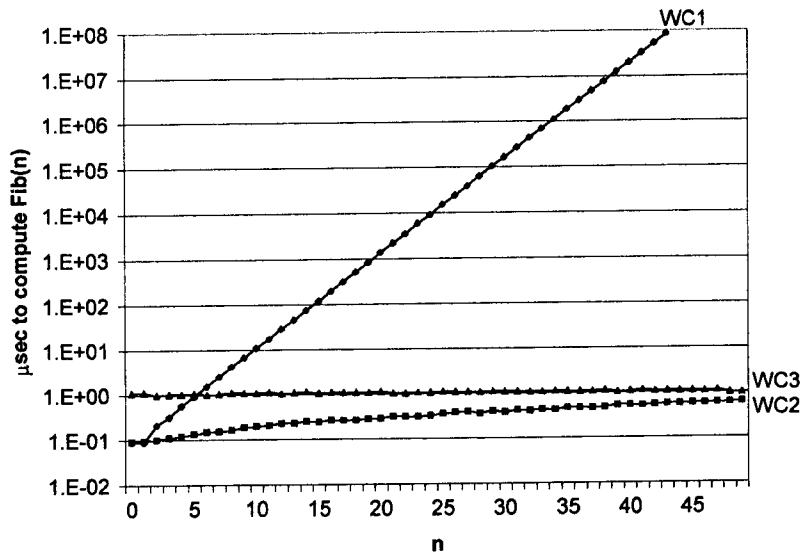
SUN workstation, C Timings (SCn)



SUN workstation, Java Timings (SJn)



Windows C Timings (WCn)



Windows Java Timings (WJn)

