PRG – Lab first partial exam

ETSInf - Academic year 2012/2013 April 22nd, 2013. Duration: 1 hour

FULL NAME AND GROUP OF LAB:

1. 2.5 points Given the following algorithm studied in lab practises for solving the Hanoi problem:

```
public static void hanoi( int n, char origin, char destination, char temporary )
{
    if ( n == 1 )
        System.out.println( "Move disk from " + origin + " to " + destination );
    else {
        hanoi( n-1, origin, temporary, destination );
        System.out.println( "Move disk from " + origin + " to " + destination );
        hanoi( n-1, temporary, destination, origin );
    }
}
```

Answer the following questions:

- If for moving 4 disk from origin to destination the algorithm does 15 movements.
  How many movements will it do for moving 5 disks?
- What should be the initial call to the algorithm if we would move 12 disks from the needle `'z'` to the `'h'` by using `'s'` as temporary?

2. 2.5 points What changes would you make in the following method to correct it?

The method `prefix( String, String )` is already implemented and runs rightly.

```
/** Returns true iff 'a' is a substring of 'b'.
 *  @param a. String.
 *  @param b. The other String.
 *  @return boolean: true iff 'a' is substring of 'b'. */
public static boolean isSubstring( String a, String b ) {
    if ( a.length() <= b.length() )
        return prefix( a, b ) || isSubstring( a, b.substring(1) );
}
```

NOTE: remind, `s.substring(int)` returns a new `String` with the characters of `s` from the position given as parameter up to the end.

3. 2.5 points The output of the fit command given the time measurements of an algorithm are the following:

```
gnuplot> f(x)=a*x+b
gnuplot> fit f(x) "algoritmo.out" using 1:2 via a,b
...
Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 0.599844         +/- 0.005358     (0.8932%)
b               = 1.00006          +/- 3324         (3.324e+05%)
```

where the input size was the length of an array. Given these results, could you give us an estimation of the running time (in nanoseconds) if the algorithm is executed with an array of 10^8 elements?

4. 2.5 points We measured the running time for the insertion-sort algorithm in the average case by means of the following code:

```
public static void insertionSortMeasurements() {
    int[] a;         // Array of integers to be sorted
    int t, r;        // Integer variables for input size and number of repetitions
    long aTime1, aTime2, aTime;   // Time
    // Print the header
    System.out.printf("# Insertion Sort \n");
    System.out.printf("#  InputSize    Average (microseconds)\n");
    System.out.printf("#------------------------\n");
    // Do the measurement process for different sizes
    for( t=INITIALSIZE; t <= MAXSIZE; t+=INCRSIZE ) {
        // Create and fill up the array with random values
        a = new int[t];
        randomArray(a);
        aTime = 0;                       // Total time set to zero initially
        for(r=0; r<REPETITIONS; r++) {
           aTime1 = System.nanoTime();   // Timestamp in nanoseconds before the call
           MeasurableAlgorithms.insertionSort( a );
           aTime2 = System.nanoTime();   // Timestamp in nanoseconds after the call
           aTime+=(aTime2-aTime1);       // Update of the total time
        }
        aTime = aTime/REPETITIONS;       // Average time
        // Print results
        System.out.printf( "%8d  %10.3f\n", t, aTime/1000.0 );
    }
}
```

The output is the following:

```
# Insertion Sort
#  InputSize  Average (microseconds)
#--------------------
   10000      44,247
   20000      80,929
   30000     123,136
   40000     166,019
   50000     209,868
   60000     254,777
   70000     300,706
   80000     347,435
   90000     395,428
  100000     443,917
```

Are these results the correct ones for the average case? Is there any error in the code? How would you correct it?