PRG – Recovering Lab First Partial Exam

ETSInf – Academic Year 2012/2013 – June 17th, 2013 – Duration: 50 minutes

NAME AND LAB GROUP:

1. 2 points Given the following algorithm studied in lab practises:

```
public static void hanoi( int n, char src, char tgt, char tmp ) {
    if (n==1) System.out.println( "Move disc from " + src + " to " + tgt );
    else {
        hanoi( n-1, src, tmp, tgt );
        System.out.println( "Move disc from " + src + " to " + tgt );
        hanoi( n-1, tmp, tgt, src );
    }
}
```

Answer the following questions:

(a) It's known that moving 10 discs from one needle to another takes 10 ms. How many milliseconds will it take for 11 discs?

(b) And for 12?

Solution:

(a) $T_{hanoi}(n) \in \Theta(2^n)$ so $T(n) = k \times 2^n$

For $n = 10$ we have $\boxed{T(10) = k \times 2^{10} = 10\ ms}$ so $\boxed{k = 10 \times 2^{-10}}$,

then $\boxed{T(11) = (10 \times 2^{-10}) \times 2^{11} = 10 \times 2 = 20\ ms}$

(b) $\boxed{T(12) = k \times 2^{12} = (10 \times 2^{-10}) \times 2^{12} = 10 \times 4 = 40\ ms}$

2. 3 points Write a **recursive** method with the following profile:

```
public static boolean isSuffix( String a, String b )
```

for checking if `a` is suffix of `b`. A string of chars is suffix of another one if all its chars appear in the same order at the end of the other string. An empty string is suffix of any other one.

NOTE: In the solution of this problem you can only use the following methods of the class `String`:

- `s.charAt(i)` returns the char stored at position `i` in `s`.
- `s.substring(i,j)` returns the substring of `s` with the chars of `s` from position `i` up to `j-1`.
- `s.length()` returns the length of `s`.

Solution:

```
public static boolean isSuffix( String a, String b ) {
      if ( a.length() == 0 ) return true;
      else if ( a.length() > b.length() ) return false;
      else return a.charAt( a.length()-1 ) == b.charAt( b.length()-1 ) &&
                  isSuffix( a.substring( 0, a.length()-1 ),
                            b.substring( 0, b.length()-1) );
}
```

3. [5 points] Given the following table with the running time of two algorithms expressed in milliseconds:

# Talla	Alg. A	Alg. B
5000	28.403	18.526
10000	112.171	75.771
15000	252.470	160.113
20000	448.906	289.250
25000	701.659	447.238
30000	1010.749	667.121
35000	1375.908	914.553
40000	1797.765	1160.216
45000	2277.439	1523.284
50000	2806.755	1835.431

Answer the following questions:

(a) What is the asymptotic temporal cost that best fits the data in column `Alg. A`?

(b) What is the asymptotic temporal cost that best fits the data in column `Alg. B`?

(c) Write an estimation of the time in **seconds** needed by both algorithms for an input size of 150000.

(d) Which algorithm has better asymptotic behaviour? Which one would you use in real applications?

Solution:

(a) The typical function that best fits the data in column `Alg. A` is n^2, so $T_A(n) \in \Theta(n^2)$.

If we focus our attention in $T_A(n)$ and $T_A(2n)$ we can see that the time for $2n$ is four times the time for n, from n^2 to 2^2n^2. Taking real measures $T_A(20000) = 448.906$ and $T_A(40000) = 1797.765$, we can see that $T_A(40000) \approx 4 \times T_A(20000)$.

(b) The typical function that best fits the data in column `Alg. B` is also n^2, so $T_B(n) \in \Theta(n^2)$. Again, if we take real measures, $T_B(20000) = 289.250$ and $T_B(40000) = 1160.216$, we can see again that $T_B(40000) \approx 4 \times T_B(20000)$.

(c) $T_A(150000) \approx 10^2 \times 252.470 \approx 25000\ milliseconds = 25\ seconds$

$T_B(150000) \approx 10^2 \times 160.113 \approx 16000\ milliseconds = 16\ seconds$

(d) Both algorithms have the same asymptotic behaviour, but $T_B(n)$ increases slower than $T_A(n)$. Therefore, if we have available two algorithms with the same asymptotic behaviour we have to focus our attention in the multiplicative factor. The factor of the algorithm B is lower than the factor of the algorithm A. In this case we will choose the algorithm B for using it in a real application.