CS 3364 - Summer I/2004 - Test 1

June 14, 2004

Name: _____

Password: _____

1 Big-O

On the following set of equations, please state for each line if $f(x) = O(g(x))$ or $g(x) = O(f(x))$. Either one, both, or none may be true. (5 pts, $\frac{1}{2}$ pt each)

$f(x)$	$g(x)$	$f(x) = O(g(x))$	$g(x) = O(f(x))$
1	x		
x	x^2		
$5x^3 + 6x^2 + 3x + 14$	$17x^3 + 4x^2 + \log x$		
x^2	x^3		
$x!$	$x^{987654321}$		
$\log x$	$\sqrt{x}$		
$\log x$	$\ln x$		
$x + x \log x + \log x$	x^2		
$x + \log x! + logx$	x^2		
$123456789 \log x$	$7x$		

2 Recurrence Relations

Bring the following Recurrence Relations into a closed form. You don't need to simplify it, but there must be no recursion left. For all exercises you may assume that $n \geq 1$ and n is an integer. *Please show your work. A one-line-answer will not count!* (5 pts, 1 pt each)

$$f(n) = \begin{cases} 0 & \text{for } n = 1 \\ f(n-1) + 1 & \text{for } n > 1 \end{cases}$$

You may assume that n is a power of 2:

$$f(n) = \begin{cases} 1 & \text{for } n = 1 \\ 2f(\frac{n}{2}) + 1 & \text{for } n > 1 \end{cases}$$

You may assume that n is even:

$$f(n) = \begin{cases} 1 & \text{for } n \leq 2 \\ f(n-2) + n & \text{for } n > 1 \end{cases}$$

You may assume that n is a power of 2:

$$f(n) = \begin{cases} 1 & \text{for } n = 1 \\ f(\frac{n}{2}) + 2n + 1 & \text{for } n > 1 \end{cases}$$

You may assume that n is a power of 3:

$$f(n) = \begin{cases} 2 & \text{for } n = 1 \\ 4f(\frac{n}{3}) + 2n - 1 & \text{for } n > 1 \end{cases}$$

3 Analysis

3.1 Array

given the following codesnippet, what will be the contents of the array ar

- after the first loop? (0 pt)
- after the second loop? (1 pt)

```
integer ar[1..10]

...

for i = 1 to 10 do
  ar[i] = i
od

for i = 1 to 10 do
  ar[i] = ar[(i+4)\%10+1]
od
```

index	1	2	3	4	5	6	7	8	9	10
1st loop										
2nd loop										

3.2 A recursive function

The following questions refer to the following code snippet, given in pseudocode. Please ask if the notation is unclear

```
integer ar[1..1000]

...

integer function dosomething(integer a, integer b)

if a == b then return ar[a]

integer c = dosomething(a,(a+b) DIV 2)
integer d = dosomething((a+b) DIV 2+1,b)

if (c < d) return c
else return d
```

Given that the array *ar* is filled with 1000 random elements, what does `dosomething(1,1000)` actually do? (1 pt)

Please be very careful on the next two questions, since both index and data are of type `integer`
Please give the reccurrence relation for index comparisons (1 pt)

$$i(n) = \begin{cases} & \text{for} \\ & \\ & \text{for} \end{cases}$$

Please give the reccurrence relation for data comparisons (1 pt)

$$d(n) = \begin{cases} & \text{for} \\ & \\ & \text{for} \end{cases}$$

3.3 Fibonacci

The following function computes the so-called Fibonacci Function:

```
function fibonacci(integer n)
if n <= 2 return 1
else return fibonacci(n-1) + fibonacci(n-2)
```

Please give the reccurrence realtion for the number of additions (1 pt)

$$a(n) = \begin{cases} & \text{for } n = 1 \\ & \text{for } n = 2 \\ & \text{for } n \geq 3 \end{cases}$$

The following non-recursive function is a more efficient way of computing the value of the fibonacci function:

```
int fib(int n)
    {
    int a = 1, b = 1;
    for (int i = 3; i <= n; i++) {
        int c = a + b;
        a = b;
        b = c;
    }
    return b;
    }
```

Please give the number of additions (1 pt)

for $n = 1$: $a(1) =$

for $n = 2$: $a(2) =$

for $n \geq 3$: $a(n) =$

3.4 Mergesort

While Mergesort has a similar performance on 1023 elements compared to 1024, Mergesort will be much slower with 1025 elements than with 1024 elements. Why (approx. 3 - 7 sentences)? (2 pt)