Algorithmic Analysis

Problem 1

Consider the following program with two unspecified lines.

```
for j = 1 to n :
  (*)
  while i > 1 :
    print i
    (**)
  end while
end for
```

Give an asymptotic upper bound on the running time, in terms of n for the given program when the missing lines are specified as follows:

(a) $(*) : i = n \qquad (**) : i = i - 1$

(b) $(*) : i = n \qquad (**) : i = i/2$

(c) $(*) : i = j \qquad (**) : i = i - 2$

(d) $(*) : i = j \qquad (**) : i = i/2$

Problem 2

Analyse the complexity of the following algorithms to compute the n-th Fibonacci number

(a) **FibOne**(n):

if $n \leq 2$ then return 1

else return **FibOne**$(n-1)$ + **FibOne**$(n-2)$

(b) **FibTwo**(n):

$x = 1$, $y = 0$, $i = 1$

While $i < n$:

  $t = x$

  $x = x + y$

  $y = t$

  $i = i + 1$

return x

† indicates a previous exam question

* indicates a difficult/advanced question.

Problem 3

Analyse the complexity of the following recursive algorithm to test whether a number x occurs in an *ordered* list $L = [x_1, x_2, \dots, x_n]$ of size n. Take the cost to be the number of list element comparison operations.

BinarySearch$(x, L = [x_1, x_2, \dots, x_n])$:

if $n = 0$ then return no

else

if $x_{\lceil \frac{n}{2} \rceil} > x$ then return **BinarySearch**$(x, [x_1, \dots, x_{\lceil \frac{n}{2} \rceil - 1}])$

else if $x_{\lceil \frac{n}{2} \rceil} < x$ return **BinarySearch**$(x, [x_{\lceil \frac{n}{2} \rceil + 1}, \dots, x_n])$

else return yes