written assignment 1

Jhao Ting Chen, Jui Li

July 2018

1 Question 1 (a)

Algorithm:

$fib(n):$
if $n \leq 0$ **then**
  return 0
else if n=1 **or** n=2 **then**
  return 1
else
  if n *is odd* **then**
    $k \leftarrow (n+1)/2$
    return $fib(k) * fib(k) + fib(k-1) * fib(k-1)$
  else
    $k \leftarrow n/2$
    return $2 * fib(k-1) + fib(k) * fib(k)$

2 Question 1 (b)

Yes, because $\mathcal{O}(\log_2 n) \leq \mathcal{O}(n^k)$ and the output memory is equal to $\log n$ bits, then all the running time is equal to $\mathcal{O}(\log_2 n + \log n) \leq \mathcal{O}(n^k)$, so we can compute F_n in time that is bounded by a polynomial.

3 Question 1 (bonus)

$ax^{17} + bx^{16} + 1 = 0$, $x^2 - x - 1 = 0$
let $F_n = x^n$ represent n^{th} Fibonacci sequence
$\Rightarrow x^{n+2} = F_{n+1} * x + F_n$ has the simmilarity with $F_{n+2} = F_{n+1} + F_n$
$\Rightarrow a(F_{17}x + F16) + b(F_{16}x + F_{15}) + 1 = 0$
$\Rightarrow (aF_{17} + bF_{16})x + aF_{16} + bF_{15} + 1 = 0$
$\Rightarrow aF_{17+}bF_{16} = 0$, $aF_{16} + bF_{15} + 1 = 0$
$a = 987$

4 Question 2 (a)

$T(A) = 0.1n^2 \log n = \mathcal{O}(n^2)$
$c = 0.1 + 1 = 1.1$
$n0 = 0$
$gA(n) = n^2 \log n$

$T(B) = 2.5n^2 = \mathcal{O}(n^2)$
$c = 2.5 + 1 = 3.5$
$n0 = 0$
$gB(n) = n^2$

if $n \leq 10$ **then**
  $gA(n) \leq gB(n)$
else
  $gA(n) > gB(n)$
therefore, when$n < 10^9$
if $n < 10$, it's better to use Algorithm A.
if $n \geq 10$, Algorithm B is better.

5 Question 2 (b)

for $f \in \mathcal{O}(g)$:
If $f \in \mathcal{O}(g)$, then $f(n) \leq c * g(n)$
$f(n) = \log n^{\log n} \rightarrow \log \rightarrow f(n) = \log n * \log n$
$g(n) = 2^{\log_2 n^2} \rightarrow \log \rightarrow g(n) = \log_2 n * \log_2 n * \log 2$
in every case when $n \geq 1$,$f(n) \leq c * g(n)$
therefore, $f \in \mathcal{O}(g)$

for $f \in \Omega(g)$:
If $f \in \Omega(g)$, then $f(n) \geq c * g(n)$
$f(n) = \log n^{\log n} \rightarrow \log \rightarrow f(n) = \log n * \log n$
$g(n) = 2^{\log_2 n^2} \rightarrow \log \rightarrow g(n) = \log_2 n * \log_2 n * \log 2$
However, only when $n \leq 1$ that $\log n \geq \log_2 n$
And, n is always ≥ 1
therefore, $f \notin \Omega(g)$

6 Question 2 (c)

If $\mathcal{O}(f+g) \in \mathcal{O}(max(f,g))$, then $f + g \leq c * max(f,g)$
for $n \geq 1$, $f(n) \leq max(f,g)$...(1)
for $n \geq 1$, $g(n) \leq max(f,g)$...(2)

(1) + (2) $f(n) + g(n) \leq 2 * max(f, g)$ (c=2)
therefore, $\mathcal{O}(f + g) \in \mathcal{O}(max(f, g))$