CS 3511: Algorithms Honors, Homework 1

Instructor: Prof. Merrick Furst
TAs: Karthik Gopalakrishnan, Pranathi Reddy Tupakula, Jiachen Li

Please submit a LaTeX-formatted PDF containing your solutions on T-Square, by Feb 13, 3:00 pm.

1 Complexity

1. (5 points) For each pair of functions f and g, write whether f is in $O(g)$, $\Omega(g)$, or $\Theta(g)$.
   (a) $f = (n + 1000)^4$, $g = n^4 - 3n^3$
   (b) $f = \log_{1000} n$, $g = \log_2 n$
   (c) $f = n^{1000}$, $g = n^2$
   (d) $f = 2^n$, $g = n!$
   (e) $f = n^n$, $g = n!$
2. (5 points) Prove that Big-O is transitive by relation. That is, if $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
3. (5 points) Determine the Big-O time complexity for the algorithm below (**show your work**). Also, very briefly explain (in one or two sentences) what the algorithm outputs (note: the % symbol is the modulo operator):

```
Data: n
i = 1
while i ≤ n do
    j = 0
    k = i
    while k % 3 == 0 do
        k = k/3
        j++
    end
    print i,j
    i++
end
```

4. (10 points) Prove that

$$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$$

(For the curious, this is useful when analyzing online caching algorithms.)

2 Master Theorem

You are trying to choose between the following three algorithms to solve a problem:

1. Algorithm A solves the problem by dividing it into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.

2. Algorithm B solves the problem of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.

3. Algorithm C solves the problem of size n by dividing it into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

(8 points) What are the asymptotic running times (in big-O notation) of each of these algorithms? Which one would you choose?

3 Identities about Fibonacci Numbers

1. (5 points) Show that
$$F_0 + F_1 + \cdots + F_n = F_{n+2} - 1$$

2. (5 points) Show that
$$F_{n-1} \cdot F_{n+1} = F_n^2 + (-1)^n$$

3. (10 points) Let $\phi = \frac{1+\sqrt{5}}{2}$ and $\overline{\phi} = \frac{1-\sqrt{5}}{2}$. Show that:
$$F_n = \frac{\phi^n - \overline{\phi}^n}{\sqrt{5}}$$
(Hint: Remember we saw in class that ϕ and $\overline{\phi}$ were the solutions to the equation $x^2 - x - 1 = 0$, albeit in a slightly different context?)

4 Modular Arithmetic

1. (2 points) Show that if $a \equiv b \pmod{N}$ and M divides N, then $a \equiv b \pmod{M}$

2. (5 points) Calculate $2^{125} \bmod 127$ using any method you choose.

3. (5 points) The least common multiple (lcm) of two integers x and y is the smallest number divisible by both x and y. Prove that $gcd(x, y) \cdot lcm(x, y) = x \cdot y$, where $gcd(x, y)$ is the greatest common divisor of x and y.

4. (5 points) Give an efficient algorithm to compute the least common multiple of two n-bit numbers x and y. What is the running time of your algorithm as a function of n?