CMPT 307 : Quiz 1 (Total Marks 55) Time: 50 minutes

1. (10 points) Rank the following functions by order of growth: that is find an arrangement $g_1, g_2, \ldots, g_6$ of the functions satisfying $g_1 \in \Omega(g_2), g_2 \in \Omega(g_3), \ldots, g_5 \in \Omega(g_6)$.

$n^2, n!, \log_e(n!), n^{0.1}, 2\log_2 n, n \log_2 n$

Answer: We first simplify :

- $\log_e(n!) \in \Theta(n \log_e n)$ i.e. $\log_e(n!) \in \Theta(n \log_2 n)$.
- $2^{\log_2 n} = n$. Note that $2^{\log_2 n} = n^{1/2}$, since $\log_2 n = \frac{\log_{10} n}{\log_{10} 2} = \frac{1}{2}$ (why?).

The functions in non-increasing asymptotic order are:

$g_1 = n!, g_2 = n^2, g_3 = n \log_2 n, g_4 = \log_e(n!), g_5 = n, g_6 = n^{0.1}$

Note that $g_3 \in \Theta(g_4)$.

2. (5 points) Suppose $\text{mystery}(n)$ is a function call taking $O(\sqrt{n})$ time to consider. Consider

```plaintext
if (\text{mystery}(n))
    A;
else
    B;
```

Give a tight bound on the running time of this piece of code as a function of $n$, on the assumption that

(a) A and B take $O(n)$ and $O(1)$ times respectively.
(b) A and B both take $O(1)$ time.

Answer: Here the function $\text{mystery}(n)$ with running time $O(\sqrt{n})$ returns a boolean value true or false.

In case (a), in the worst case, the running time of $A$ dominates the running time of $B$. The total running time cost is, therefore, $O(n + \sqrt{n})$ which is $O(n)$ (why?).

In case (b), in the worst case, the running time is dominated by that of $\text{mystery}(n)$. Therefore, the running time is $O(\sqrt{n})$. 

1
3. (5 points) Assume the parameter $n$ in the procedure below is a positive power of 2, i.e. $n = 2, 4, 8, 16, \ldots$. Determine the running time of the following function.

Function mystery(n)
{
    count = 0; x = 2;
    while (x < n ) {
        x = 2 * x;
        count++
    }
    writeln(count)
}

**Answer:** Suppose $n = 2^k$ for some $k$. In this case $k = \log_2 n$. The initial value of $x$ is 2. Every iteration of the while loop doubles the value of $x$. Hence the while loop will be repeated $k - 1$ times. Therefore, the total step count is $O(k)$ which is $O(\log_2 n)$.

4. (15 points) This question is on modular arithmetic.

(a) Starting from the definition of $x \equiv y \pmod{N}$ (namely, $N$ divides $x - y$), show that

$$x \equiv x' \pmod{N}, \ y \equiv y' \pmod{N} \ \Rightarrow \ \ xy \equiv x'y' \pmod{N}$$

**Answer:** Since $N$ divides $x - x'$ and $y - y'$, $x = x' + tN$ and $y = y' + t'N$ where $t$ and $t'$ are integers. Therefore, $xy = (x' + tN) * (y' + t'N) = x'y' + N(t'x' + ty' + t'tN)$. Therefore, $xx' - yy'$ is divisible by $N$. This implies $xx' = yy' \pmod{N}$.

(b) Show that if $a \equiv 1 \pmod{N}$, then $a^p \equiv 1 \pmod{N}$

**Answer:** This is obtained by multiplying $a \equiv 1 \pmod{N}$ repeatedly $p$ number of times.

(c) Give a polynomial time algorithm to compute $x^y \pmod{N}$ where $x, y, N$ are all $n$-bit positive integers. You must analyze your algorithm for the worst-case time complexity.

**Answer:** Discussed in the text (Figure 1.4).
5. (10 points) Describe Euclid’s algorithm to compute the greatest common
divisor of a positive integers $M$ and $N$ of $n$ binary bits long. Show that the
algorithm runs in polynomial time.

**Answer:** Euclid’s algorithm is described in Figure 1.5 of the text. The algorithm is a recursive algorithm. The lemma in page 21 says that
if $M \geq N \Rightarrow M \mod N < \frac{M}{2}$.

Let $r_1$ be the remainder, and $r_1 < M/2$.

In the second round, using the same logic, we can claim that the remainder $r_2 (= N \mod r_1)$ will be less than $N/2$.

This means that in two consecutive rounds, both the arguments $M$ and $N$
are at least halved in their values. The length of each argument decreases by at least one bit. Therefore, the base case will be reached in $2n$ recursive
calls (initially, $M$ and $N$ are $n$-bit long). Each recursive call requires a mod computation. Therefore, the worst case complexity of the gcd algorithm is $O(n^3)$ which is a polynomial in $n$.

6. (10 points) This question is about Fermat’s Little Theorem (FLT).

(a) Formally state FLT.

(b) What is the contrapositive equivalent of FLT?

(c) Use FLT to show that $8^{62} \equiv 8^2 \pmod{11}$.

**Answer:**

(a) (Text page 23) If $p$ is prime, then $\forall 1 \leq a < p, a^{p-1} \equiv 1 \pmod{p}$.

(b) The contrapositive equivalent: if $\exists 1 \leq a < p, a^{p-1} \not\equiv 1 \pmod{p}$, then
$p$ is not a prime.

(c) We can write $8^{62} = 8^2 \times 8^{60} = 8^2 \times (8^{10})^6$. FLT says that $8^{10} \equiv 1 \pmod{11}$. Therefore, $8^{62} \equiv 8^2 \times (8^{10})^6 \equiv 8^2 \times (8^{10} \equiv 1 \pmod{11})^6 \equiv 8^2 \pmod{11}$. (mod 11).