A. Lubiw, B. Ma, E. Zima

ASSIGNMENT 1

DUE: Wednesday September 20, 7 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies. **Note:** All logarithms are base 2 (i.e., $\log x$ is defined as $\log_2 x$).

1. [12 marks] **Order notation.** For each of the following pairs of functions $f(n)$ and $g(n)$, determine the "most appropriate" symbol in the set $\{O, o, \Theta\}$ to complete the statement that $f(n) \in \quad (g(n))$ (if one of the symbols applies at all). "Most appropriate" means that you should not answer "O" if you could answer "o" or "Θ". Justify your answers.

   You may use the following (based on Skeina, p. 56), where $f(n) \ll g(n)$ is shorthand for $f(n) \in o(g(n)$:

   $$1 \ll \log\log n \ll \log n \ll \log^2 n \ll \sqrt{n} \ll n \ll n\log n \ll n^2 \ll 2^n \ll n!$$

   Furthermore, $n^a \in o(n^b)$ for $0 < a < b$, and $\log^a n \in o(n^b)$ for any $b > 0$, and $n^a \in o(2^n)$.

   (a) $f(n) = 2017n^3 + 12871n^2 + 19$, $g(n) = \frac{2}{2017}n^4 + 2n$;
   (b) $f(n) = \log^2(n^4)$, $g(n) = \sqrt{n}$;
   (c) $f(n) = 16^{\log n^3}$, $g(n) = \frac{1}{3}n^{12}$;
   (d) $f(n) = n^2$, $g(n) = (\lceil \frac{n}{2} \rceil - \frac{n}{2})n^2$;

2. [10 marks] **Analysis of Run Time.** Analyze the following pseudocode and give a tight (Θ) bound on the running time as a function of n. You can assume that all individual instructions are elementary, i.e., take constant time. Show your work.

```
m := n; l:=0; s:=0;
while m > 1 do
    for j = n − m to n do
    |   l := l+1
    end
    for j = 1 to ⌊log n⌋ do
    |   s := s + 1
    end
    m := m/2
end
```

3. [10 marks] **Reductions.** In class we saw an $O(n^2)$ time algorithm for the following 3-SUM problem:

   Given an array $A[1..n]$ of n numbers, find, if they exist, indices, i, j, k, $1 \le i, j, k \le n$ such that $A[i] + A[j] + A[k] = 0$.

Consider the more general problem of 3-SUM with 3 different arrays:

Given three arrays X, Y, Z where X has n_x integers, Y has n_y integers, Z has n_z integers, and $n_x + n_y + n_z = n$. Find, if they exist, indices i, j, k in the appropriate ranges such that $X[i] + Y[j] + Z[k] = 0$.

(a) [2 marks] Suppose you define array $A[1..n]$ to be the concatenation of X, Y and Z and run the old 3-SUM algorithm on A. Give an example to show that does not solve the general 3-SUM problem.

(b) [8 marks] Give a correct reduction from 3-SUM with 3 different arrays to 3-SUM. You should not give a stand-alone algorithm or alter the 3-SUM code. HINT: Consider 3 arrays $X'[i] = 10*X[i]+1$, $Y'[i] = 10*Y[i]+2$, and $Z'[i]$ defined in some similar manner that you should figure out. Argue that your method is correct and give the run time.

4. [10 marks] **Recurrences.** Some divide-and-conquer algorithms give rise to the recurrence

$$T(n) = 2T(n/2) + n \log n, \quad T(1) = 1.$$

From the Master Theorem, we know that $T(n) \in \Theta(n \log^2 n)$. The goal of this question is to prove $T(n) \in O(n \log^2 n)$ by induction without using the Master Theorem. To simplify, assume that n is a power of 2. Thus you must prove by induction on k that $T(2^k) \in O(2^k \log^2 2^k)$.