Homework 5 Rubric

CMSC 203

1. Section 3.1, Exercise 14 (10pts, 5pts ea)
   List all the steps (including any comparisons) used to search for 7 in the sequence $1, 3, 4, 5, 6, 8, 9, 11$ using

   a) a linear search

      i. compare 7 to 1
      ii. compare 7 to 3
      iii. compare 7 to 4
      iv. compare 7 to 5
      v. compare 7 to 6
      vi. compare 7 to 8
      vii. compare 7 to 9
      viii. compare 7 to 11

   b) a binary search

      i. find the middle element of {1, 3, 4, 5, 6, 8, 9, 11}, rounding down (5)
      ii. compare 7 to 5
      iii. reduce the list to the upper half of the list: {6, 8, 9, 11}
      iv. find the middle element of {6, 8, 9, 11}, rounding down (8)
      v. compare 7 to 8
      vi. reduce the list to the lower half of the list, excluding 8: {6}
      vii. compare 7 to 6

2. Section 3.1, Exercise 24 (6pts)
   Describe an algorithm that determines whether a function from a finite set of integers to another finite set of integers is one-to-one.
   One extremely inefficient algorithm would be to find the mapping for the first element of the domain, then compare it with the mappings of every other element in the domain and see if there were any conflicting mappings. Then repeat this process for every element in the domain.
   A more efficient method would be to store the mappings of every element in the domain in a list and use a binary search to check if an element was already contained in the list.
   An even better algorithm would be to construct the hashmap for every element in the domain

3. Section 3.1, Exercise 34 (10pts)
   Use the bubble sort to sort $6, 2, 3, 1, 5, 4$, showing the lists obtained at each step.

   a) $\{2, 6, 3, 1, 5, 4\}$
   b) $\{2, 3, 6, 1, 5, 4\}$
   c) $\{2, 3, 1, 6, 5, 4\}$
   d) $\{2, 3, 1, 5, 6, 4\}$
   e) $\{2, 3, 1, 5, 4, 6\}$
   f) $\{2, 3, 1, 5, 4, 6\}$

g) $\{2, 1, 3, 5, 4, 6\}$
h) $\{2, 1, 3, 5, 4, 6\}$
i) $\{2, 1, 3, 4, 5, 6\}$
j) $\{1, 2, 3, 4, 5, 6\}$
k) $\{1, 2, 3, 4, 5, 6\}$
l) $\{1, 2, 3, 4, 5, 6\}$
m) $\{1, 2, 3, 4, 5, 6\}$
n) $\{1, 2, 3, 4, 5, 6\}$
o) $\{1, 2, 3, 4, 5, 6\}$

4. Section 3.1, Exercise 38 (10pts)
Use the insertion sort to sort the list in Exercise 34, showing the lists obtained at each step.

a) $\{6, 2, 3, 1, 5, 4\}$
b) $\{2, 6, 3, 1, 5, 4\}$
c) $\{2, 6, 3, 1, 5, 4\}$
d) $\{3, 2, 6, 1, 5, 4\}$
e) $\{2, 3, 6, 1, 5, 4\}$
f) $\{2, 3, 6, 1, 5, 4\}$
g) $\{1, 2, 3, 6, 5, 4\}$
h) $\{1, 2, 3, 6, 5, 4\}$
i) $\{5, 1, 2, 3, 6, 4\}$
j) $\{1, 5, 2, 3, 6, 4\}$
k) $\{1, 2, 5, 3, 6, 4\}$
l) $\{1, 2, 3, 5, 6, 4\}$
m) $\{1, 2, 3, 5, 6, 4\}$
n) $\{4, 1, 2, 3, 5, 6\}$
o) $\{1, 4, 2, 3, 5, 6\}$
p) $\{1, 2, 4, 3, 5, 6\}$
q) $\{1, 2, 3, 4, 5, 6\}$
r) $\{1, 2, 3, 4, 5, 6\}$

5. Section 3.2, Exercise 2 (12pts, 2 pts ea)
Determine whether each of these functions is $O(x^2)$.

a) $f(x) = 17x + 11$
Yes

b) $f(x) = x^2 + 1000$
Yes

c) $f(x) = x \log x$
Yes

d) $f(x) = x^4/2$
No

e) $f(x) = 2^x$
No

f) $f(x) = \lfloor x \rfloor \cdot \lceil x \rceil$
Yes

6. Section 3.2, Exercise 4 (10pts)
Use the definition of "$f(x)$ is $O(g(x))$" to show that $2^x + 17$ is $O(3^x)$. Include the witness c and k.
Let $c = 1$ and $k = 3$. $2^3 + 17 = 25$ and $3^3 = 27$. 3^x will triple every time x increases, wheras $2^x + 17$ will at most double every time x increase. Therefore $2^x + 17$ is $O(3^x)$ with $c = 1$ and $k = 3$ as witnesses.

7. Section 3.2, Exercise 14abce (8pts, 2 pts ea)
Determine whether x^3 is $O(g(x))$ for each of these functions $g(x)$.

a) $g(x) = x^2$
No

b) $g(x) = x^3$
Yes

c) $g(x) = x^2 + x^3$
Yes

e) $g(x) = 3^x$
Yes

8. Section 3.2, Exercise 26 (9pts, 3 pts ea)
Give a big-O estimate for each of these functions. For the function g in your estimate $f(x)$ is $O(g(x))$, use a simple function g of smallest order.

a) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
$n^3 \log n$

b) $(2^n + n^2)(n^3 + 3^n)$
$2^n 3^n$

c) $(n^n + n2^n + 5^n)(n! + 5^n)$
$n^n n!$

9. Section 3.3, Exercise 12a (10pts)
Consider the following algorithm , which takes as input a sequence of n integers $a_1, a_2, \ldots, a_n$ and produces as output a matrix $\mathbf{M} = \{m_{ij}\}$ where m_{ij} is the minimum term in the sequences of integers $a_i, a_{i+1}, \ldots, a_j$ for $j \geq i$ and $m_{ij} = 0$ otherwise.

initialize $\mathbf{M}$ so that $m_{ij} = a_i$ if $j \geq i$ and $m_{ij} = 0$ otherwise
for $i := 1 \to n$ **do**
  for $j := i + 1 \to n$ **do**
    for $k := i + 1 \to j$ **do**
      $m_{ij} := \min(m_{ij}, a_k)$
return $\mathbf{M} = \{m_{ij}\}\{m_{ij}$ is the minimum term of $a_i, a_{i+1}, \ldots, a_j\}$

a) Show that this algorithm uses $O(n^3)$ comparisons to compute the matrix $\mathbf{M}$.
The innermost assignment will take constant time to do the comparison and assignment.
The innermost loop (with k as the index) will run no more than n times. [The minimum possible value to initialize k is $i + 1$, and the minimum possible value for i is 1 and the maximum value k will ever go to is j and the maximum value that j will go to is n.]
The middle loop (with j as the index) will run no more than n times. [The minimum possible value to initialize j is $i + 1$ which is 2. The maximum value j will go to is n]
The outermost loop (with i as the index) will run n times.
Therefore the runtime of the nested loops will take $O(n) \cdot O(n) \cdot O(n)$ time
The initialization step of $\mathbf{M}$ will do one assignment for every element in $\mathbf{M}$. $\mathbf{M}$ has n^2 elements, so this step will take $O(n^2)$ time.
Therefore the total runtime will be $O(n^3 + n^2) = O(n^3)$.

10. Chapter 3, Supplementary Exercise 26 (15pts)
Arrange the functions $2^{100n}, 2^{n^2}, 2^{n!}, 2^{2^n}, n^{\log n}, n \log n \log \log n, n^{3/2}, n(\log n)^{3/2},$ and $n^{4/3}(\log n)^2$ in a list so that each function is big-O of the next function. [*Hint:* To determine the relative size of some of these functions, take logarithms.]

a) $n \log n \log \log n$
b) $n(\log n)^{3/2}$
c) $n^{4/3}(\log n)^2$
d) $n^{3/2}$
e) 2^{100n}
f) 2^{n^2}
g) 2^{2^n}
h) $2^{n!}$
i) $n^{\log n}$