Sample Exam #3

1. Write the pseudocode for an algorithm that takes a list of n integers $a_1, a_2, ..., a_n$ and finds the number of integers each greater than five in the list.

   Solution:

   procedure $greaterthanfive(a_1, ..., a_n : integers)$

   answer := 0

   for $i := 1$ **to** n

       if $a_i > 5$ **then** $answer := answer + 1$

   return $answer$

2. Write the pseudocode for a brute-force algorithm that finds the largest product of two numbers in a list $a_1, a_2, ..., a_n$ $(n \geq 2)$ that is less than a threshold N.

   Solution:

   procedure $largestproduct(a_1, a_2, ..., a_n, N : realnumbers)$

   $largestproduct := -\infty$

   for $i = 2$ **to** n

       for $j = 1$ **to** $i - 1$

           if $a_i \cdot a_j < N$ **then**

           if $a_i \cdot a_j > largestproduct$ **then** $largestproduct := a_i \cdot a_j$

   return $largestproduct$ $\{largestproduct$ is the largest product of two numbers in the list that is less than N, or $-\infty$ if all products are greater than or equal to $N\}$

3. List all the steps used to search for 26 in the list: 4, 6, 7, 13, 16, 20, 26, 31 using

   (a) a linear search

   Solution: Compare 26 to the first entry in the list, $26 \neq 4$, so compare 26 with the second entry in the list, $26 \neq 6$, so compare 26 with the third entry in the list, $26 \neq 7$ so continue comparing and moving the the next entry until you get to the seventh entry in the list where $26 = 26$, a match is found and the location "7" is returned.

   (b) a binary search

   Solution: To search for 26 in the list:

   $$4, 6, 7, 13, 16, 20, 26, 31$$

   first split this list, which has 8 terms, into two smaller lists with 4 terms each, namely,

   $$4, 6, 7, 13 \quad 16, 20, 26, 31.$$

Then, compare 26 and the largest term in the first list. Because $13 < 26$, the search for 26 can be restricted to the list containing the 5th through the 8th terms of the original list. Next, split this list, which has 4 terms into the two smaller lists of two terms each, namely,

$$16, 20 \quad 26, 31.$$

Because $20 < 26$ (comparing 26 with the largest term of the first list) the search is restricted to the second of these lists, which contains the 7th and 8th terms of the original list. The list is split into to lists, namely,

$$26 \quad 31.$$

Since 26 is not less than 26 the search is restricted to the first of these two lists, which only contains the 7th term. Since the list is narrowed down to one term only a comparison is made and 26 is located as the 7th term of the original list.

4. Use the bubble sort algorithm to sort 31, 16, 7, 6, 20, show the lists obtained at each step.

   Solution:

   First pass: $[31, 16, 7, 6, 20]$, $[16, 31, 7, 6, 20]$, $[16, 7, 31, 6, 20]$, $[16, 7, 6, 31, 20]$, $[16, 7, 6, 20, 31]$.

   Second pass: $[16, 7, 6, 20, 31]$, $[7, 16, 6, 20, 31]$, $[7, 6, 16, 20, 31]$, $[7, 6, 16, 20, 31]$.

   Third pass: $[7, 6, 16, 20, 31]$, $[6, 7, 16, 20, 31]$, $[6, 7, 16, 20, 31]$.

   Fourth pass: $[6, 7, 16, 20, 31]$.

5. Use the definition of big$-O$ to prove that $1^3 + 2^3 + \cdots + n^3$ is $O(n^4)$.

   Solution:

$$1^3 + 2^3 + \cdots + n^3 \leq n^3 + n^3 + \cdots + n^3 = n \cdot n^3 = n^4$$

6. Let $f(n) = 3n^2 + 8n + 7$. Show that $f(n)$ is $O(n^2)$. Find the witnesses C and k from the definition.

   Solution:

   $f(n) \leq 3n^2 + 8n^2 + 7n^2 = 18n^2$ if $n \geq 1$; therefore $C = 18$ and $k = 1$ can be used.

7. Prove that $\dfrac{x^3 + 7x^2 + 3}{2x + 1}$ is $\Theta(x^2)$

   Solution:

   $\dfrac{x^3 + 7x^2 + 3}{2x + 1}$ is $O(x^2)$ since, if $x \geq 1$

$$\frac{x^3 + 7x^2 + 3}{2x + 1} \leq \frac{x^3 + 7x^3 + 3x^3}{2x} = \frac{11x^3}{2x} = \frac{11}{2}x^2.$$

Also, x^2 is $O(\frac{x^3+7x^2+3}{2x+1})$ since, if $x \geq 1$

$$x^2 = \frac{x^3}{x} \leq \frac{x^3+7x}{2x} \leq \frac{x^3+7x^2+3}{2x+1}.$$

8. Find all pairs of functions in this list that are of the same order: $n^2 + \log(n)$, $2^n + 3^n$, $100n^3 + n^2$, $n^2 + 2^n$, $n^2 + n^3$, $3n^3 + 2^n$.

   Solution:

$$(100n^3 + n^2,\ n^2 + n^3),\ (3n^3 + 2^n, n^2 + 2^n).$$

9. Suppose you have two different algorithms for solving a problem. To solve a problem of size n, the first algorithm uses exactly $n\sqrt{n}$ operations and the second algorithm uses exactly $n^2 \log(n)$ operations. As n grows, which algorithm uses fewer operations?

   Solution: The first algorithm uses fewer operations as n grows.

10. For the following questions, find the best big-O notation to describe the complexity of the algorithm. Choose your answers from the following:

$$1, \log_2(n), n, n\log_2(n), n^2, n^3, ..., 2^n, n!.$$

   (a) An algorithm that prints all subsets of size three of the set $\{1, 2, 3, ..., n\}$.
   Solution: n^3

   (b) The number of print statements in the following:
   $i := 1$, $j := 1$ **while** $i \leq n$
       while $j \leq i$
           print hello;
           $j := j + 1$
       $i := i + 1$
   Solution: n^2

   (c) A linear search to find the smallest number in a list of n numbers.
   Solution: n

11. Give a big-O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm:

   $t := 0$
       for $i = 1$ **to** n
           for $j = 1$ **to** n
             $t := (it + jt + 1)^2$

   Solution: $O(n^2)$