MAT2440/D648 **Exam #3**
Instructor: Suman Ganguli **Due: Thursday, May 18** **Name:** _____

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
Total:	50	

1. (10 points) Consider a function "`sort`" which takes as input a list of 5 integers (i.e., input $(a_0, a_1, a_2, a_3, a_4)$ where each $a_i \in \mathbb{Z}$), and returns the list sorted in ascending order. For example:

$$\texttt{sort}(9, 4, 0, 5, -1) = (-1, 0, 4, 5, 9)$$

   (a) What is the domain of `sort`? Express the domain as a Cartesian product.

   (b) Show that `sort` is *not* a one-to-one function.

2. (10 points) We analyzed that the worst-case time complexity of linear search is $O(n)$ while the time complexity of binary search is $O(\log n)$.

   (a) What does the variable n represent here?

   (b) Briefly explain what aspect of the binary search algorithm makes its time complexity $O(\log n)$. (It may be helpful to do #2 before answering this question–included on the next page is the pseudocode for binary search.)

   (c) Based on their big-O estimates, which of these search algorithms is preferable to use for large values of n? Why?

3. (10 points) Here is pseudocode which implements binary search:

procedure `binary-search` (x : integer, $a_1, a_2, \ldots a_n$: increasing integers)
  $i := 1$ (the left endpoint of the search interval)
  $j := n$ (the right endpoint of the search interval)
  while $(i < j)$:
    $m := \left\lfloor \frac{i+j}{2} \right\rfloor$
    if $(x > a_m)$ **then**: $i := m + 1$
    else: $j := m$
  if $(x = a_i)$ **then**: location := i
  else: location := 0
  return location

Fill in the steps used by this implementation of binary search to find the location of $x = 38$ in the list

$$a_1 = 17, a_2 = 22, a_3 = 25, a_4 = 38, a_5 = 40, a_6 = 42, a_7 = 46, a_8 = 54, a_9 = 59, a_{10} = 61$$

- **Step 1:** Initially $i = 1, j = 10$ so search interval is the entire list

  $$a_1 = 17, a_2 = 22, a_3 = 25, a_4 = 38, a_5 = 40, a_6 = 42, a_7 = 46, a_8 = 54, a_9 = 59, a_{10} = 61$$

- **Step 2:** Since $i = 1 < j = 10$, the algorithm enters the **while** loop. Using the values of i and j:

  $m =$ _____ and so $a_m =$ _____
  From comparing x and a_m, the updated values of i and j are

  $i =$ _____ and $j =$ _____
  and so the new search interval is the sublist: _____

- **Step 3:** Since $i < j$, the algorithm again enters the **while** loop again. Using the current values of i and j:

  $m =$ _____ and so $a_m =$ _____
  From comparing x and a_m, the updated values of i and j are

  $i =$ _____ and $j =$ _____
  and so the new search interval is the sublist: _____

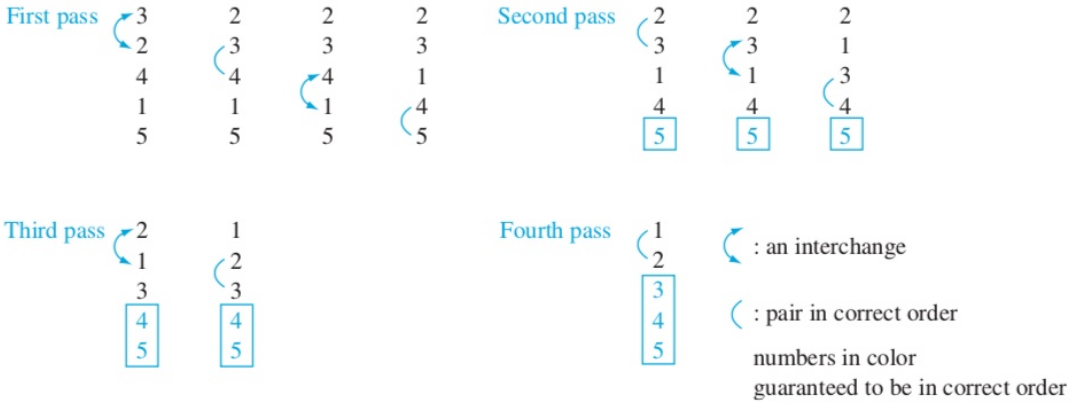
- **Step 4:** Since $i < j$, the algorithm again enters the **while** loop again. Using the current values of i and j:

  $m =$ _____ and so $a_m =$ _____
  From comparing x and a_m, the updated values of i and j are

  $i =$ _____ and $j =$ _____
  and so the new search space is the sublist: _____

- **Step 5:** Since $i = j$, the algorithm does not enter the while loop. What does the algorithm do then, and what value does it return?

4. (10 points) Shown is an example of using bubble sort to sort a list of integers:



(a) Briefly describe the bubble sort algorithm. You can use the example and the pseudocode (see the textbook or the slides) as a guide, but you should **describe how the algorithm works in general** (in your own words), on any given list of numbers. (Hint: Describe how many "passes" are made through the list, and what the algorithm does on each pass.)

(b) Use bubble sort to put the following list into alphabetical order, showing the lists obtained at each step of the algorithm (as in the example above).
Note that since there are 4 elements in the list, bubble sort requires $4 - 1 = 3$ passes:

First pass: Second pass:

`NY`

`NJ`

`PA`

`CT`

Third pass: Sorted list:

5. (10 points) Review the proof of the following theorem by mathematical induction (as presented in class and in the textbook, as Example 1 in Section 5.1):

Theorem: For any positive integer n,

$$1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

Fill in the steps in the proof of this theorem:

Proof (by induction):For any given positive integer n, we will use $P(n)$ to represent the proposition:

$$P(n) : 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

Thus, we need to prove that $P(n)$ is true for $n = 1, 2, 3, ...$, i.e., we need to prove:

$$(\forall n \in \mathbb{N})P(n)$$

For a proof by mathematical induction, we must prove the base case (namely, that $P(1)$ is true), and we must prove the inductive step, i.e., that the conditional statement

$$P(k) \longrightarrow P(k+1)$$

is true, for any given $k \in\in \mathbb{N}$.

(a) **Base case:** Show that the base case $P(1)$ is true:

(b) **Inductive step:** In order to provide a direct proof of the conditional $P(k) \longrightarrow P(k+1)$, we start by assuming $P(k)$ is true, i.e., we assume

$$1 + 2 + 3 + \ldots + k = \frac{k(k+1)}{2}$$

Now use this assumption to show that then $P(k+1)$ is true.
(Hint: note that the the proposition $P(k+1)$ is the equation:

$$1 + 2 + 3 + \ldots + k + (k+1) = \frac{(k+1)((k+1)+1)}{2}$$

Start with the LHS of this equation, and show that it is equal to the RHS, using the assumption/equation $P(k)$!)