CMSC 351 Midterm I Exam Spring 2019

Name (PRINTED): ________________________________

Student ID #: ________________________________

This exam is closed-book, closed-notes, and closed-calculators, except you may use the handout. Show your work. *Clarity and neatness count.*

If you need more space, you can use the blank space in the back. Make sure to cross reference it.

There are six questions. Good luck.

1 (10):
2 (15):
3 (15):
4 (30):
5 (30):

1. (10 points) Answer the following questions briefly, using the following list of numbers, $[15, 5, 10, 3, 8, 4, 6, 1]$:

   (a) How would the list look like after three complete passes of insertion sort without sentinel?

   Solution:
   $[3, 5, 10, 15, 8, 4, 6, 1]$

   (b) How would the list look like after three complete passes of selection sort?

   Solution:
   $[1, 5, 6, 3, 4, 8, 10, 15]$

2. (15 points)

   (a) At level, j, of a recursion tree for merge sort algorithm, What is the size of a subproblem? How many subproblems are there? How many comparisons are carried out at that level?

   Solution:
   $\frac{n}{2^j}$, 2^j, $2^j(\frac{n}{2^j} - 1)$

   (b) How many comparisons (exact numeric number) would the merge routine use to merge the following two sorted arrays, $[2, 3, 8, 12]$ and $[4, 5, 6]$?

   Solution: 5

3. (15 points) Using the array, $A = [6, 5, 4, 3, 2, 1]$ as an input to the following algorithm:

```
for i = n downto 2 do
    for j = 1 to i-1 do
        if A[j] > A[j+1] then A[j] ↔ A[j+1]
    end for
end for
```

   (a) What is the exact number of comparisons carried out to sort this array in the increasing order?

   Solution: $\frac{n(n-1)}{2} = \frac{6\times5}{2} = 15$

   (b) Justify your answer in 2-3 sentences how you found the answer in Part(a).

   Solution: It is a bubble sort algorithm and the worst case number of comparisons is $\frac{n(n-1)}{2}$

4. (30 points) Find the maximum and minimum elements in a given list of, n, numbers. Assume n is even.

   (a) Write pseudocode for a brute force algorithm. Analyze it to find the exact number of comparisons?

Solution:

```
min = A[1]
for i = 2 to n:
    if min > A[i]:
        min = A[i]
```

The code will need $n - 1$ comparisons to find min.
Run it again to find the max for a total of $n - 1 + n - 1 = 2n - 2$ comparisons.

   (b) Write pseudocode for an optimized algorithm. Analyze it to find the exact number of comparisons?

Solution:

```
if A[0] > A[1]:
    max = A[0]
    min = A[1]
else:
    max = A[1]
    min = A[0]
while i <= n-1
    if A[i] < A[i+1]:
        if A[i] < min:
            min = A[i]
        if A[i+1] > max:
            max = A[i+1]
    else:
        if A[i] > max:
            max = A[i]
        if A[i+1] < min:
            min = A[i+1]
    i = i + 2
```

Exact number of comparisons:

$$\sum_{i=2}^{\frac{n}{2}} 3 = 3[\frac{n}{2} - 1] + 1$$
$$= \frac{3n}{2} - 2$$

5. (30 points) We are given an array of n positive integers and a target sum, x, and we want to find a subarray whose sum is x or report that there is no such subarray.

   (a) Write pseudocode for a brute force algorithm. What is its runtime?

Solution:

```
for i = 1 to n - 1:
    S = A[i]
    for j = i + 1 to n:
        if S == x:
            return S[i:j-1]
        if S > x:
            break out of for j loop
        S = S + A[j]
return -1,-1
```

Runtime: $\theta(n^2)$

(b) Write pseudocode for an optimized algorithm. What is its runtime?

Solution:

```
S = 0
i = 1
j = 2
while i <= n:
    while i < j-1 and S > x:
        S = S - A[i]
        i += 1
    if S == x:
        return i,j-1
    if j < n:
        S = S + A[j]
    j = j + 1
return -1,-1
```

Runtime: $\theta(n)$