NJIT New Jersey Institute of Technology

Number on the list	Grade

Midterm Exam 1

Time: 2 hours

Do not remove the top-left staple. Your papers must remain attached during the exam.

Please write your name once you get seated.

Name:

Instruction:

1. Smart devices (cellphones, smart watches, AirPods, ...) are not allowed during the exam. Please ensure that they are turned off and left your cellphones at the designated area before the exam.
2. Under no circumstances are students allowed to leave the exam room during the exam, for any reason.
3. The exam is closed book, and using any study materials, laptops, notes, or textbooks during the exam is prohibited.
4. Backpacks and other personal belongings (like jacket, laptop, ...) should be placed under your desk before the exam starts. Accessing your backpack during the exam is not allowed, so please make sure you have all necessary items on your desk before the exam begins.
5. Please check that the number on this paper matches your assigned number on the list, and that you sit on the correct seat.
6. You are not allowed to use your own paper during the exam. If you need extra paper, please ask the instructor.
7. If you erase or cross out any part of your solution and then rewrite it, the rewritten solution will not be eligible for regrading if needed. Please ensure your final answers are clear and unaltered to facilitate accurate grading.

1. a) Determine the time complexity of the following algorithm. hint: $\sum_{k=0}^{m} 2k+1 = (m+1)^2$ (15 points).

```
s=0
k=0
While s<n do
    s=s+2k+1
    k=k+1
```

b) solve the following recurrence relation: (15 points).
T(n)=3T(n/3)+1, T(1)=1

Hint: $\sum_{k=0}^{m} a^k = \frac{a^{m}-1}{a-1}$

a)
$s = 0, k = 0$
$s = 1, k = 1$
$s = 1 + 3, k = 2$
$s = 1 + 3 + 5, k = 3$
$s = 1 + 3 + 5 + 7, k = 4$
:

$$s = \sum_{k=0}^{m} 2k + 1 = (m+1)^2 \approx m^2$$

$s = n \Rightarrow m^2 = n \Rightarrow m = \sqrt{n} \Rightarrow T(n) = O(\sqrt{n})$

b)

$$T(n) = 3T\left(\frac{n}{3}\right) + 1$$

$$T(n) = 3\left(3T\left(\frac{n}{9}\right) + 1\right) + 1 = 9T\left(\frac{n}{9}\right) + 3 + 1 = 9\left(3T\left(\frac{n}{27}\right) + 1\right) + 3 + 1 = 27T\left(\frac{n}{27}\right) + (9 + 3 + 1)$$
$$= 27\left(3T\left(\frac{n}{81}\right) + 1\right) + (9 + 3 + 1) = 81T\left(\frac{n}{81}\right) + (27 + 9 + 3 + 1)$$

$$T(n) = 3^k T\left(\frac{n}{3^k}\right) + (3^{k-1} + \cdots + 3 + 1) = 3^k T\left(\frac{n}{3^k}\right) + \left(\frac{3^k - 1}{3 - 1}\right)$$

Setting $3^k = n$, we have:

$$T(n) = nT(1) + \left(\frac{n-1}{3-1}\right) = n + \left(\frac{n-1}{2}\right) = O(n)$$

2. a) Explain in place heapsort. (10 points)
b) Given a list of n numbers, provide an efficient algorithm that returns true if there is a duplicate, and false otherwise, without using extra space. (15 ponits)

b)

```
Algorithm has_duplicate(arr):
    arr.heapsort()  # Sort the array in-place using heapsort in O(n log n)
    for i=0 to len(arr) – 1 do #O(n)
        if arr[i] == arr[i + 1] then
            return True
    return False
```

time complexity is O(nlog n) and we don't use extra space

3. Apply Bubble Sort to the following list and pause after 3 iterations (when the length of the sorted part is 3). (10 points)

4	2	9	3	1	5	6

b) Modify Bubble_sort to stop once the list is sorted. (15 points)

a) first iteration

4	2	9	3	1	5	6

2	4	9	3	1	5	6

2	4	3	9	1	5	6

2	4	3	1	9	5	6

2	4	3	1	5	9	6

2	4	3	1	5	6	9

Second iteration:

2	3	4	1	5	6	9

2	3	1	4	5	6	9

Third iteration:

2	3	1	4	5	6	9

b)

```
for i = n-1 down to 1 do
   flag = False
  for  j = 0 to i-1 do
       if  A[j] > A[j+1] then
            swap(A[j], A[j+1])
            flag = True
if  flag == False then
          break
```

4. a) You are given an algorithm and its input size. Determine the time complexity of each algorithm from the following options (A to F). (5 points)
$A: O(n), B: O(n^2 \log^2 n), C: O(n^2), D: O(nlog\ n),\ E: O(n^2 \log n), F: O(\sqrt{n})$

Algorithm	Input size	Time complexity
Heapsort	n^2	
Bucket sort	$nlog\ n$	
Counting sort	$n/2$	
Bubble sort	$n \log n$	
quicksort	$\sqrt{n}$	

b) You are a data analyst in a department store. You have access to *n* different discount percentages over a period of time (one discount for each week). Some weeks may have no discount, but the discount is a number between 0 and 75%, with at most one decimal point (e.g., 12.5%, 25%, or 5.5%). Write pseudocode to sort them in linear time. What is the number of spaces you use? (15 points)

a) E, D, A, B, A

b) The array will have 760 elements to account for all the possible discount values from 0% to 75%, in increments of 0.1%. We use counting sort by placing each discount d at the index 10 * d in the count array.

```
function mysort(D):

    # Step 1: Create a count array for 760 possible discount values (0.0% to 75.0%)

    C = array of 760 zeros

    for i=0 to n do

       index = D[i] * 10  # Multiply by 10 to handle decimal points

       C[index] += 1

    # Step 3: Reconstruct the sorted list

    D = []

    for i =0 to 759 do

        while C[i] > 0 do

            D.append(i / 10.0)

            C[i] -= 1

        return D
```

we use n cells for input and output and 760 cells for storing the count of each item, overall O(n) spaces.