Graded Assignment 2

Q1)

```
def fun(n):
  s = 0
  for i in range(0, n):
    for j in range(0, n):
      for k in range(0, n):
        s += 1
  for i in range(0, n):
    for j in range(0, n):
      s -= 1
  for i in range(0, n):
    s -= 1
  return s
```

What is the time complexity of the given function fun ?

A1) $O(n^3)$

Q2) Let B(n),A(n) and W(n) be best-case,average-case,and worst-case running time of an algorithm. executed on an input size **n**. Which of the following is/are always **True**.

A2)

- A(n) = O(W(n))
- A(n) = Ω(B(n))
- B(n) = O(W(n))
- B(n) = O(A(n))

Q3) What is the asymptotic complexity of merge sort when the input is sorted in reverse order?

A3) O(nlogn)

Q4)

```
def selectionsort(L):
  n = len(L)
  if n < 1:
    return(L)
  for i in range(n):
    mpos = i
    for j in range(i + 1, n):
      if L[j] < L[mpos]
        mpos = j
    (L[i], L[mpos]) = (L[mpos], L[i])
  return(L)
```

What is the value of **i** when the list [9,4,5,2,3,7,6,8,1] becomes completely sorted for the first time?

A4) **5**

Selection sort swaps min element to i = 0th position element and i = 1th position with the second minimum element from the list and so on by doing up to i = 5 we get completely sorted list.

Q5)

```
def insertionsort(L):
  n = len(L)
  if n < 1:
    return(L)
  for i in range(n):
     j = i
     while(j > 0 and L[j] < L[j - 1]):
      (L[j], L[j - 1]) = (L[j - 1], L[j])
      j = j - 1
  return(L)
```

What of the following statement(s) is /are correct with regard to the given insertion sort? [MSQ]

A5)

- ☑ The sort is stable and it sorts in-place
- ☑ After m iterations of the for-loop, the first m elements in the list are in sorted order.

Q6) A program is written in 3 stages where the first stage is of O(nlogn), the second stage is of $O(n^2)$ which is based on the result of the first stage,and the third stage is of O(n). WHat will be asymptotic complexity of the entire program?

A6) **O(n^2)**

Time complexity will be O(nlogn + n^2 + n) since we take higher-order term and can neglect lower order term for a large value of n, Hence O(n^2) is correct complexity.

Q7) A school wants to maintain a databse of its students. Each student has a unique id and it is stored along with other details. Adding a new student with a unique id, searching for a student using their id, and removal of students are the frequent operation performed on the database. From the options given below,choose the most efficient technique to store the data.

A7)

- ☑ Maintain a sorted list with id. Whenever a new student is added, insert the student details into the respective positon in the sorted list by id.

Q8)

```
def tsearch(L, x):
   global c
   c += 1
   n = len(L)

   if n == 0:
    return False
   if L[n // 3] == x:
    return True
   if L[2 * n // 3] == x:
    return True

   if x < L[n // 3]:
    return tsearch(L[:n // 3], x)
   elif x > L[2 * n // 3]:
    return tsearch(L[2 * n // 3:], x)
   else:
    return tsearch(L[n // 3 : 2 * n // 3], x)
```

Choose the order of complexity of the search function **tsearch.**

A8) O(logn)

Q9) Which of the following statement(s) is/are true?

I: $(n+3)^k = \Omega(n^k)$

II: $n^x = \theta(n^y)$, if and only if $x = y$

A9) I and II both are true.

Q10) Arrange the following functions in increasing order of asymptotic complexity

- f1(n) = 3n + logn
- f2(n) = $(\log n)^2$
- f3(n) = log(logn)
- f4(n) = 100logn
- f5(n) = 3nlogn

  A10)

  f3(n),f4(n),f2(n),f1(n),f5(n)

  f3 < f4 < f2 < f1 < f5 by putting n = 10^{31} we can verify this.
