Question 1 (20 points): Solve the recurrence relations below.

a. (5 points): $T(n) = T(n-1) + n$

b. (5 points): $T(n) = T(\alpha n) + T((1-\alpha)n) + cn \quad (0 < \alpha < 1)$

c. (5 points): $T(n) = 4T(n/2) + n^3$

d. (5 points): $T(n) = 2T(n/4) + \lg n$

Question 2 (10 points**)** Describe the Mergesort and Quicksort (not the randomized version) algorithms, and explain the similarity and difference between both algorithms. You should give the algorithm details like how the merging function or the partition function works.

Question 3 (10 points**)** What are the minimum and maximum numbers of elements in a heap of height *h*? Explain your answer.

Question 4 (10 points**)** Show that the running time of Quicksort (not the randomized version) is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

Question 5 (10 points) Explain why any comparison sort algorithm requires $\Omega(nlgn)$ comparisons in the worst case.

Question 6 (10 points) Read the algorithm of counting sort below, and answer the following questions

```
COUNTING-SORT(A, B, k)

1 let C[0 .. k] be a new array
2 for i = 0 to k
3       C[i] = 0
4 for j = 1 to A.length
5       C[A[j]] = C[A[j]]+1
6 for i = 1 to k
7       C[i] = C[i] + C[i-1]
8 for j = A.length down to 1
9       B[C[A[j]]] = A[j]
10      C[A[j]] = C[A[j]]-1
```

a. (7 points) Suppose that we were to rewrite the for loop header in line 8 of the COUNTING-SORT as **for** *j* = A.*length* **to** 1. Show that the algorithm still works properly. Is the modified algorithm stable?

b. (3 points) Is it a good idea to use counting sort to sort a set of n integers in the range of [1, n^3]? Please explain your answer.

Question 7 (10 points) Given an array A of n elements. Write an $\Theta(n)$ algorithm to find the minimum, maximum, and the i-th largest elements from A.

Question 8 (10 points) Write an algorithm to find the 'next' node (i.e., in-order successor) of a given node in a binary search tree where each node has a link to its parent.

Question 9 (10 points) Given an English word as an input, describe an algorithm that output all valid anagrams for the input word. An anagram of a word is a rearrangement of the letters in that word, so that it forms another English word. For example, “stop” is the anagram of “post”. (Hint: use a hash table)

Question 10 (10 points) Write an algorithm to find whether a graph has a cycle.

Question 11 (10 points) Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut. Show that the converse is not true by giving a counterexample.

Question 12 (10 points): Tell whether each of (a) - (e) is TRUE, FALSE, or OPEN.

(i) $\Pi_1 \in P$,

(ii) $\Pi_2 \in$ NP-complete,

(iii) $\Pi_3 \in$ NP.

---

a. $\Pi_1 \in$ NP

b. $\Pi_2 \in$ P

c. Π_2 is polynomially reducible to Π_1

d. If P = NP, then $\Pi_2 \notin$ P

e. Π_3 is polynomially reducible to Π_2