Q1:

(10 points) Let G = (V, E) be an undirected graph, connected, or each edge has a non-negative weight. Here is a strategy divide-and-reign to the problem of finding a minimum spanning tree for this graph:

We partition the set of vertices V into two sets V1 and V2 such that their cardinality differs at most 1. E1 is the set of edges that are obiter at the vertices V1 and E2 is the set of edges that are obiter at the vertices V2. We then recursively solves the problem of finding a minimum spanning tree for each of the sub-graphs G1 = (V1, E1) and G2 = (V2, E2). We then recursively solves the problem of finding a minimum spanning tree for each of the sub-graphs G1 = (V1, E1) and G2 = (V2, E2).

Prove that this algorithm computes a minimum spanning tree well for G or give a against-example.

Q2:

(10 points) Given the sequence, a sequence is a sub sequence of if there is a strictly increasing sequence of indices of such that. For example, is a subsequence of Corresponding to the sequence of indices

Given the 2 sequences and we say that sequence Z is a sub-sequence common to and if is a subsequence of and a subsequence of . For Example, if and , is a subsequence in common of X and Y and in fact is the longest sequence in common of and .

1. (10 pts) Give a dynamic programming algorithm that calculates the longest common subsequence between two sequences and . What is the complexity of the algorithm?
2. (5 pts) Build a dynamic Programming Table for your algorithm in
3. A palindrome is a string of non-blank characters that reads the same way from left to right and right to left (for example: (palindromes phobia), radar, dwell, etc..). Write an algorithm for

Q3:

Let S and T, any two sequences over a finite alphabet Σ of size | S | = n and | T | = m. How many possible alignments between S and T if the operations are considered only identities (match), substitutions (mismatch) and deletions (i.e. is not considered insertions). (Hint: The resulting alignment will be "-" only one of the two sequences.)

Q4:

(10 points) Let P (length m) pattern and T (of length n), m << n a circular genome (see Figure 1). Describe an algorithm in O (n ^ 2 m) to obtain all instances approximate positions (with at most k errors) of P in T.

For example, if P = AAGT, T=GATAAGTAA and k = 1, there is an exact match of P starting at position 4 and ending at position 8 of the text and a case with an error starting at position Enlarge 8 and ending at position 3 (circular occurrence).

Q5:

(20 points) Let A [1 .. n] an array of n distinct integers. If i <j and A [i]> A [j], we say that the pair (i, j) is an inversion of A.

1. (5pts) What is the maximum number of inversions can have an array A of size n? Justify your answer.
2. (5pts) What is the relationship between the execution time of insertion sort and the number of inversions in the input array? Justify your answer
3. (10pts) Write an algorithm whose time complexity in the worst case is Θ (n log n), which determines the number of inversions presented in any permutation of n elements. (Hint: change sort-merge)

Q6:

(20 points) Let be a list of elements ranges in an array of indices to . It is assumed that the only operation that we know perform on the Elements is to check if two elements are equal or not. We say that element is dominant if all has strictly more than elements. Here it is assumed that is a power of .

1. (10pts) Give a naive algorithm to compute the number of elements of the set Equals \_x for x gives. Deduce an algorithm to check if L possesses a dominant element. What is the time complexity in the worst case your algorithm?
2. Give another algorithm divide-and-conquer to this problem. What is the time complexity in the worst case your algorithm?