ASSIGNMENT II

1. Write an algorithm to find the two closest points (using Euclidean distance) in the plane by brute force method. Analyze its Time complexity.
2. *Alternating disks:* You have a row of 2*n* disks of two colours, *n* dark and *n* light. They alternate: dark, light, dark, light, and so on. You want to get all the dark disks to the right-hand end, and all the light disks to the left-hand end. The only moves you are allowed to make are those that interchange the positions of two neighbouring disks.



Determine the number of moves it takes.

1. Consider the problem of counting, in a given text, the number of substrings that start with an A

and end with a B. For example, there are four such substrings in CABAAXBYA. Design a brute-

force algorithm for this problem and determine its efficiency class.

1. Write pseudocode for a divide-and-conquer algorithm for finding the position of the largest

element in an array of *n* numbers. Set up and solve a recurrence relation for the number of key

comparisons made by your algorithm.

1. Apply quicksort to sort the list A, N, P, A, N, M, A, N in alphabetical order. Draw the tree of the recursive calls made.
2. The following algorithm seeks to compute the number of leaves in a binary tree.

**ALGORITHM** *LeafCounter(T )*

//Computes recursively the number of leaves in a binary tree

//Input: A binary tree *T*

//Output: The number of leaves in *T*

**if** T = ∅ **return** 0

**else return** LeafCounter(Tlef t)+ LeafCounter(Tright)

Is this algorithm correct? If it is, prove it; if it is not, make an appropriate correction.

Apply the above/corrected algorithm to count the number of nodes in the following tree.

