

CS3230

Tutorial 5

1. Give an algorithm which finds the distance between closest pair of points, when all the points lie on a straight line (though not necessarily all on the x axis). Try to make the algorithm simpler than the one done in class.
2. Suppose we are sorting an array consisting of integers ≥ 0 . Explain how one may modify the inputs (in linear time) so that any sorting algorithm becomes a stable sorting algorithm.

Your modification of the numbers should be such that one can easily obtain the original numbers back from the modification.

3. Suppose A is a sorted array of n **distinct** integers. Give an algorithm to find if there exists an i such that $A[i] = i$ (and in case such an i exists, then return one such i).

For example, if the array contains $-4, -2, 2, 4, 7$, then $A[4] = 4$.

One can find such an i by just checking whether $A[j] = j$, for each j between 1 and n (both inclusive). However this is $O(n)$ algorithm.

Can you give a better algorithm? Give time complexity (worst case) analysis of your algorithm.

4. Suppose A is adjacency matrix of a simple graph. Show that $A^m[i, j]$ gives number of paths from i to j of length exactly m

(here A^0 is the identity matrix (which has 1 in the diagonals and 0 everywhere else), $A^1 = A$, and $A^{n+1} = A^n * A$).

5. Consider the following coin denominations. For each part, say whether the greedy algorithm done in class gives an optimal answer. If so, give an argument to justify your answer. If not, then give a counterexample.

- (a) 1, 5, 7
- (b) 1, 5, 15
- (c) 1, 4, 9
- (d) 1, 3, 5