

Assignment 3

Due Date: Before the class on September 19.

Some Important Points

1. Make a group of at most 2 people. The group members remain same till the end of this course.
2. Collaboration is encouraged. However, each group should write its final answer separately.
3. Please mention names of all the people with whom you have discussed the question.
4. Please try to make at least one attempt at the question without using the hints.

1. (10 points) Suppose you are given an array $A[1 \dots n]$ such that each element of A is either *white* or *black*. Design an $O(n)$ time algorithm to sort this array such that all the white elements are listed before all the black elements. Your algorithm should take only $O(1)$ extra memory (over the array A which is given to you). Prove that your algorithm is correct.

Hint: Can anything done in the QuickSort algorithm be of use here?

2. (10 points) Assume that an unsorted array A contains n distinct numbers. An inversion in A is a pair of elements with index i and j such that $A[i] > A[j]$ but $i < j$. Design an algorithm that finds the total number of inversions in A in $O(n \log n)$ time. Prove that your algorithm is correct.

Hint: Will a modified mergesort be of any help here?

3. (10 points) An array $A[1 \dots n]$ having distinct numbers is *bitonic* if there exists an $1 \leq i \leq n$ such that $A[1] < A[2] < \dots < A[i-1] < A[i] > A[i+1] \dots A[n-1] > A[n]$. Design an $O(n)$ time algorithm to sort this array. Prove the correctness of your algorithm. Also assume that you don't know the range of these n numbers (so an algorithm like **Radix sort** is ruled out).

4. (10 points) Prove that any algorithm that finds an element x in the sorted list of n elements will take $\approx \log n$ time.

Hint: Change the proof discussed in the class which shows that sorting n elements takes $\approx n \log n$ time.

5. (10 points) In **QuickSort**, instead of picking the first element as the **pivot**, assume that you always pick the middle element as the pivot. What is the running time this modified QuickSort on an already sorted array?