

DSA TUTORIAL 5

SORTING ALGORITHMS

DIFFERENT TYPES OF SORTS -

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Bucket Sort
- Counting Sort
- sort() in C++

IMPLEMENTATIONS:

- [Implementation of Merge Sort](#)
- [Implementation of Quick Sort](#)
- [Implementation of sort\(\) in C++ \(using custom compare function\).](#)

PROBLEMS:

- [Count Inversions in an array](#)

Given an array, find out how many swaps are required to make the array sorted. Swaps can only be done on adjacent elements.

Sample Input -

5
2 4 1 3 5

Sample Output -

3

Explanation -

These pairs need to be swapped - (4,1), (2,1), (4, 3)

Solution - <http://p.ip.fi/Jvbq>

- [The Crofts Game](#)

There are 2 people, Alice and Bob. Alice has an array of integers A and Bob has an array of integers B. Both arrays are of size n. Alice and Bob are playing a game, where both play alternating turns. In each turn, the player will select an index i ($0 \leq i \leq n-1$). If Alice selects the index, then she will get A[i] points and if Bob selects the index, he will get B[i] points. The game ends when all the indices have been selected. You have to find which player has maximum points at the end of the game.

Input format:

There are t testcases. In each testcase, the input is :

First line - n - no. of integers in each array

Second line - n space separated integers - the array A

Third line - n space separated integers - the array B

Sample Input :

1

5

8 2 4 6 3

4 5 10 7 2

Sample Output :

Alice

Explanation:

Round 1 : Alice picks index 0. Alice - 8, Bob - 0

Round 2 : Bob picks index 2. Alice - 8, Bob - 10

Round 3 : Alice picks index 3. Alice - 14, Bob - 10

Round 4 : Bob picks index 1. Alice - 14, Bob - 15

Round 5 : Alice picks index 4. Alice - 17, Bob - 15

Thus, Alice wins.

Note - This is one optimal solution. There can be multiple ways in which both players play optimally.

Solution - <http://p.ip.fi/jLV0>

- [A Pancake Sorting Problem](#)

Given an array of integers, you have to sort it by performing only the following operation on the array:

flip(arr, i) - flips the array from index 0 to index i

Assume that flip(arr, i) takes $O(1)$ time.

Find an efficient algorithm for sorting the array.

Sample Input:

6

10 300 20 200 30 100

Sample Output:

10 20 30 100 200 300

Solution - <http://p.ip.fi/vM-B> (**Note** : In the solution, the flip takes $O(n)$ time but don't consider it in finding the time complexity).

PRACTICE PROBLEMS:

- [Chef and Card Trick](#)
- [Descending Sort](#)
- [Merge Sort for Linked Lists](#)
- [TimSort](#)
- [Collecting packages](#)
- [Array Splitting](#)