201B172

Jaypee University of Engineering and Technology

18B11CI311– Data Structures
B.Tech -3rd Semester
Tutorial – 5 (Sorting Algorithms)

Consider the following arrays (Contains 8 elements to be sorted in ascending order) -

Array W - 4, 5, 7, 1, 6, 10, 17, 13
Array X - 20, 15, 13, 11, 4, 3, 2, 1
Array Y - 32, 4, 6, 9, 12, 3, 11, 10
Array Z - 5, 8, 12, 17, 19, 23, 41, 50

1. Give the number of swaps in 4th pass in bubble sort for array W.
2. What is the time complexity of selection sort for array X and give the output of 4th pass.
3. With reference to insertion sort, which of above arrays (W, X, Y, Z) will be worst, average and best case. Justify your answer with algorithmic statements.
4. Array Z is the worst case input instance of quick sort. Justify the statement.
5. Compare quick and merge sort. Use the above arrays (W, X, Y, Z) to give specific time complexity case.
6. What is the worst case time complexity of insertion sort where position of the data to be inserted is calculated using binary search?
7. Consider the array A[]= {6,4,8,1,3} apply the insertion sort to sort the array . Consider the cost associated with each shift is 5 rupees, what is the total cost of the insertion sort when element 1 reaches the first position of the array?
8. Let t1 and t2 be the number of comparisons made by P for the inputs {1, 2, 3, 4, 5} and {4, 1, 5, 3, 2} respectively. Mention the relation between t1 and t2 for the following cases:
   a. Quick sort using the first element as pivot.
   b. Bubble sort
   c. Insertion sort
   d. Selection sort
9. Give the practical situations where each of the following sorting algorithms should be preferred over other. Also mention which one is stable and in-place along with their time complexity (best, average and worst case).
   a. Quick sort
   b. Bubble sort
   c. Insertion sort
   d. Selection sort
   e. Merge sort
10. Apply bubble sort, selection sort, insertion sort, quick sort, and merge sort on following array. Show all the steps.
    15, 6, 2, 5, 9, 10, 45, 32, 9, 3

Date __/__/____
Page 01

Tutorial-5

Array W = 4, 5, 7, 1, 6, 10, 17, 13
Array X = 20, 15, 13, 11, 4, 3, 2, 1
Array Y = 32, 4, 6, 9, 12, 3, 11, 10.
Array Z = 5, 8, 12, 17, 19, 23, 41, 50

- following arrays (contains 8 elements to be sorted in ascending order)

(1) The number of swaps in 4th pass in bubble sort for array W is 5

for 0th pass - number of swaps = 3
and resultant array
4 5 1 7 6 10 17 13
for 1st pass - swaps = 4
array - 4 1 5 6 7 10 13 17
for 2nd pass - swaps = 5
array - 1 4 5 6 7 10 13 17
(Here array sorted)

(2) Here array X is reverse sorted so worst case time complexity is $O(n^2)$
and output of 4th pass (starting with pass 0)
→ 1, 2, 3, 4, 11, 13, 15, 20.

(3) With reference to insertion sort.

Array X → This is worst case as the array is reverse sorted.

Array Z → This is best case as the array is already sorted.

Array W and Array Y → As most of the elements are already sorted only a few swaps and passes are needed, that is why this is an average case.

(4) As array Z is already sorted which is the worst case in quick sort.

Array Z: 5, 8, 12, 17, 19, 23, 41, 50

It is already sorted and takes $O(n^2)$ time.

(6) Worst case time complexity will be $O(n^2)$ Because applying binary search to calculate the position of the data to be inserted doesn't reduce the time complexity of insertion sort. This is because insertion of a data at an appropriate position involves two steps:

(1) Calculate the position
(2) Shift the data from the position calculated in step1 one step right to create a gap where the data will be inserted.

Date __/__/____
Page 03

Insertion Sorting

(7) A[] = {6, 4, 8, 1, 3}

Step 1 ⇒ 4 6 8 1 3

Step 2 ⇒ 1 4 6 8 3

Here when the element 1 reaches the first position of the array two comparisons are only required

hence, 25*2 = 50 Rupees

(9)

Sorting	worst Case	Average Case	Best Case	Stable	in-place
Bubble sort	N^2	N^2	N	Yes	Yes
Selection sort	N^2	N^2	N^2	No	Yes
Insertion Sort	N^2	N^2	N	Yes	Yes
Quick sort	N^2	NlogN	NlogN	No	Yes
Merge sort	NlogN	NlogN	NlogN	Yes	No

Practical situations where each of the following sorting algorithms should be preferred over other:

Selection Sort: Situation when swap operation is very costly.

Insertion Sort: when the sorting applied on the array which is sorted or almost sorted.

Bubble sort: This gives fastest result on an extremely small or nearly sorted set of data.

Merge sort: Widely used for external sorting, where random access can be very, very expensive.

Quick Sort: When we don't need a stable sort and average case performance matters more than worst case performance.

(8) (b) Bubble Sort.:

$T_1 = 10$

$T_2 = 10$

$\therefore T_1 = T_2$

(c) Insertion Sort:

$T_1 = 4$

$T_2 = 8$

$T_2 > T_1$

(d) Selection Sort:

$T_1 = 10$

$T_2 = 10$

$T_1 = T_2$

(a)

Date __/__/____
Page 05

(10) Given array: 15 6 2 5 9 10 45 32 9 3

Bubble sort:

15	6	2	5	9	10	45	32	9	3
6	15	2	5	9	10	45	32	9	3
6	2	15	5	9	10	45	32	9	3
6	2	5	15	9	10	45	32	9	3
6	2	5	9	15	10	45	32	9	3
6	2	5	9	10	15	45	32	9	3
6	2	5	9	10	15	32	45	9	3
6	2	5	9	10	15	32	9	45	3
6	2	5	9	10	15	32	9	3	45
2	6	5	9	10	15	32	9	3	45
2	5	6	9	10	15	32	9	3	45
2	5	6	9	10	15	9	32	3	45
2	5	6	9	10	15	9	3	32	45
2	5	6	9	10	9	15	3	32	45
2	5	6	9	10	9	3	15	32	45
2	5	6	9	9	10	3	15	32	45
2	5	6	9	9	3	10	15	32	45
2	5	6	9	3	9	10	15	32	45
2	5	6	3	9	9	10	15	32	45
2	5	3	6	9	9	10	15	32	45
2	3	5	6	9	9	10	15	32	45

Date __/__/____
Page 06

Selection Sort:

15	6	2	5	9	10	45	32	9	3
15	6	2	5	9	10	3	32	9	45
15	6	2	5	9	10	3	9	32	45
9	6	2	5	9	10	3	15	32	45
9	6	2	5	9	3	10	15	32	45
3	6	2	5	9	9	10	15	32	45
3	5	2	6	9	9	10	15	32	45
3	2	5	6	9	9	10	15	32	45
2	3	5	6	9	9	10	15	32	45

Insertion Sort:

15	6	2	5	9	10	45	32	9	3
6	15	2	5	9	10	45	32	9	3
6	2	15	5	9	10	45	32	9	3
2	6	15	5	9	10	45	32	9	3
2	6	5	15	9	10	45	32	9	3
2	5	6	15	9	10	45	32	9	3
2	5	6	9	15	10	45	32	9	3
2	5	6	9	10	15	45	32	9	3
2	5	6	9	10	15	32	45	9	3
2	5	6	9	10	15	32	9	45	3
2	5	6	9	10	15	9	32	45	3
2	5	6	9	10	9	15	32	45	3
2	5	6	9	10	10	15	32	45	3
2	5	6	9	9	10	15	32	3	45
2	5	6	9	9	10	15	3	32	45

Date __/__/____
Page 07

2	5	6	9	9	10	3	15	32	45
2	5	6	9	9	3	10	15	32	45
2	5	6	9	3	9	10	15	32	45
2	5	6	3	9	9	10	15	32	45
2	5	3	6	9	9	10	15	32	45
2	3	5	6	9	9	10	15	32	45

Quick Sort.

15	6	2	5	9	10	45	32	9	3
15	6	2	5	3	10	45	32	9	9
3	6	2	5	15	10	45	32	9	9
3	6	2	5	9	10	45	32	9	15
3	5	2	6	9	10	45	32	9	15
3	2	5	6	9	10	45	32	9	15
2	3	5	6	9	10	45	32	9	15
2	3	5	6	9	10	45	15	9	32
2	3	5	6	9	10	9	15	45	32
2	3	5	6	9	10	9	15	32	45
2	3	5	6	9	10	15	9	32	45
2	3	5	6	9	9	15	10	32	45
2	3	5	6	9	9	10	15	32	45

Date __/__/____
Page 08

Merge Sort:

15	6	2	5	9	10	45	32	9	3
15	6	2	5	9	10	45	32	9	3
6	15	2	5	9	10	45	32	3	9
6	15	2	5	9	10	45	3	9	32
2	5	6	9	15	3	9	10	32	45
2	3	5	6	9	9	10	15	32	45

(5) For Array in Merge sort

X - It is ~~worst~~ average
W - It is average
Y - It is average
Z - It is best.

For Array in Quic sort.

X → worst
Y → avg
W → avg
Z → worst.

Case	quick sort	merge sort
best	nlogn	nlogn
avg	nlogn	nlogn
worst	n^2	nlogn