Quicksort – 1”.

1. Which of the following sorting algorithms is the fastest?
a) Merge sort
b) Quick sort
c) Insertion sort
d) Shell sort
View Answer
Answer: b
Explanation: Quick sort is the fastest known sorting algorithm because of its highly optimized inner loop.

2. Quick sort follows Divide-and-Conquer strategy.
a) True
b) False
View Answer
Answer: a
Explanation: In quick sort, the array is divided into sub-arrays and then it is sorted (divide-and-conquer strategy).

3. What is the worst case time complexity of a quick sort algorithm?
a) O(N)
b) O(N log N)
c) $O(N^2)$
d) O(log N)
View Answer
Answer: c
Explanation: The worst case performance of a quick sort algorithm is mathematically found to be $O(N^2)$.

Note: Join free Sanfoundry classes at [Telegram]() or [Youtube]()

advertisement

4. Which of the following methods is the most effective for picking the pivot element?
a) first element
b) last element
c) median-of-three partitioning
d) random element
View Answer
Answer: c
Explanation: Median-of-three partitioning is the best method for choosing an

appropriate pivot element. Picking a first, last or random element as a pivot is not much effective.

5. Find the pivot element from the given input using median-of-three partitioning method.
8, 1, 4, 9, 6, 3, 5, 2, 7, 0.
a) 8
b) 7
c) 9
d) 6
View Answer
Answer: d
Explanation: Left element=8, right element=0,
Centre=[position(left+right)/2]=6.

Take Data Structure II Practice Tests - Chapterwise!
Start the Test Now: Chapter 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

6. Which is the safest method to choose a pivot element?
a) choosing a random element as pivot
b) choosing the first element as pivot
c) choosing the last element as pivot
d) median-of-three partitioning method
View Answer
Answer: a
Explanation: This is the safest method to choose the pivot element since it is very unlikely that a random pivot would consistently provide a poor partition.

7. What is the average running time of a quick sort algorithm?
a) O(N^2)
b) O(N)
c) O(N log N)
d) O(log N)
View Answer
Answer: c
Explanation: The best case and average case analysis of a quick sort algorithm are mathematically found to be O(N log N).

8. Which of the following sorting algorithms is used along with quick sort to sort the sub arrays?
a) Merge sort
b) Shell sort
c) Insertion sort

d) Bubble sort
View Answer
Answer: c
Explanation: Insertion sort is used along with quick sort to sort the sub arrays.
It is used only at the end.

9. Quick sort uses join operation rather than merge operation.
a) true
b) false
View Answer
Answer: a
Explanation: Quick sort uses join operation since join is a faster operation than merge.

10. How many sub arrays does the quick sort algorithm divide the entire array into?
a) one
b) two
c) three
d) four
View Answer
Answer: b
Explanation: The entire array is divided into two partitions, 1st sub array containing elements less than the pivot element and 2nd sub array containing elements greater than the pivot element.

11. Which is the worst method of choosing a pivot element?
a) first element as pivot
b) last element as pivot
c) median-of-three partitioning
d) random element as pivot
View Answer
Answer: a
Explanation: Choosing the first element as pivot is the worst method because if the input is pre-sorted or in reverse order, then the pivot provides a poor partition.

12. Which among the following is the best cut-off range to perform insertion sort within a quick sort?
a) N=0-5
b) N=5-20
c) N=20-30
d) N>30
View Answer

Answer: b
Explanation: A good cut-off range is anywhere between N=5 and N=20 to avoid nasty degenerate cases.

"Quicksort – 2".

1. Quick sort is a __________
a) greedy algorithm
b) divide and conquer algorithm
c) dynamic programming algorithm
d) backtracking algorithm
View Answer
Answer: b
Explanation: Quick sort is a divide and conquer algorithm. Quick sort first partitions a large array into two smaller sub-arrays. And then recursively sorts the sub-arrays.

2. What is the worst case time complexity of the Quick sort?
a) O(nlogn)
b) O(n)
c) $O(n^3)$
d) $O(n^2)$
View Answer
Answer: d
Explanation: The worst case running time for Quick sort is $O(n^2)$. In Quick sort, the worst case behaviour occurs when the partitioning routine produces two sub-arrays one with n – 1 element and other with 0 elements.

3. Apply Quick sort on a given sequence 7 11 14 6 9 4 3 12. What is the sequence after first phase, pivot is first element?
a) 6 4 3 7 11 9 14 12
b) 6 3 4 7 9 14 11 12
c) 7 6 14 11 9 4 3 12
d) 7 6 4 3 9 14 11 12
View Answer
Answer: b
Explanation: Let's apply Quick sort on the given sequence,
For first phase, pivot = 7

```
7    11    14    6    9    4    3    12
      i                              j
```

```
7    11    14   6    9    4    3    12
      i                          j
7    3    14    6    9    4    11    12
           i                 j
7    3    4   6    9    14   11    12
              i    j
7    3    4    6   9   14    11    12
              j    i
6    3    4    7    9    14    11    12
```

4. The best case behaviour occurs for quick sort is, if partition splits the array of size n into __________
a) n/2 : (n/2) – 1
b) n/2 : n/3
c) n/4 : 3n/2
d) n/4 : 3n/4
View Answer
Answer: a
Explanation: The best case analysis of quick sort occurs when the partition splits the array into two subarrays, each of size no more than n/2 since one is of size n/2 and one of size (n/2) – 1.

5. Quick sort is a stable sorting algorithm.
a) True
b) False
View Answer
Answer: b
Explanation: In stable sorting algorithm the records with equal keys appear in the same order in the sorted sequence as they appear in the input unsorted sequence. Quick sort does not preserve the relative order of equal sort items. Therefore, Quick sort is not a stable sort.

6. Consider the Quick sort algorithm in which the partitioning procedure splits elements into two sub-arrays and each sub-array contains at least one-fourth of the elements. Let T(n) be the number of comparisons required to sort array of n elements. Then T(n)<=?
a) T(n) <= 2 T(n/4) + cn
b) T(n) <= T(n/4) + T(3n/4) + cn

c) T(n) <= 2 T(3n/4) + cn
d) T(n) <= T(n/3) + T(3n/4) + cn
View Answer
Answer: b
Explanation: If there are n/4 elements in one sub-array then T(n/4) comparisons are needed for this sub-array. And T(3n/4) comparison are required for the rest 4n/5 elements, and cn is time required for finding the pivot. If there are more than n/4 elements in one sub-array then other sub-array will have less than 3n/4 elements and time complexity will be less than T(n/4) + T(3n/4) + cn.

7. Consider the Quick sort algorithm which sorts elements in ascending order using the first element as pivot. Then which of the following input sequence will require a maximum number of comparisons when this algorithm is applied on it?
a) 22 25 56 67 89
b) 52 25 76 67 89
c) 22 25 76 67 50
d) 52 25 89 67 76
View Answer
Answer: a
Explanation: If the input sequence is already sorted then worst case behaviour occurs for the Quick sort algorithm which use the first element as pivot. Therefore, the input sequence given in 22 25 56 67 89 will require a maximum number of comparisons.

8. A machine needs a minimum of 200 sec to sort 1000 elements by Quick sort. The minimum time needed to sort 200 elements will be approximately __________
a) 60.2 sec
b) 45.54 sec
c) 31.11 sec
d) 20 sec
View Answer
Answer: c
Explanation: The Quick sort requires nlog2n comparisons in best case, where n is size of input array. So, 1000 * log21000 ≈ 9000 comparisons are required to sort 1000 elements, which takes 200 sec. To sort 200 elements minimum of 200 * log2200 ≈ 1400 comparisons are required. This will take 200 * 1400 / 9000 ≈ 31.11 sec.

9. Which one of the following sorting algorithm is best suited to sort an array of 1 million elements?
a) Bubble sort
b) Insertion sort
c) Merge sort

d) Quick sort
View Answer
Answer: d
Explanation: The Quick sort is best suited to sort the array of 1 million elements. The practical implementations of Quick sort use randomised version. In practice randomised Quick sort algorithms rarely shows worst case behaviour and is almost always O(nlogn). And Quick sort requires little additional space and exhibits good cache locality.

10. Quick sort is a space-optimised version of ____
a) Bubble sort
b) Selection sort
c) Insertion sort
d) Binary tree sort
View Answer
Answer: d
Explanation: Quick sort is a space-optimised version of the binary tree sort. In binary sort tree, the elements are inserted sequentially into the binary search tree and Quick sort organises elements into a tree that is implied by the recursive calls.

"Quicksort – 3".

1. QuickSort can be categorized into which of the following?
a) Brute Force technique
b) Divide and conquer
c) Greedy algorithm
d) Dynamic programming
View Answer
Answer: b
Explanation: First you divide(partition) the array based on the pivot element and sort accordingly.

2. Select the appropriate recursive call for QuickSort.(arr is the array, low is the starting index and high is the ending index of the array, partition returns the pivot element, we will see the code for partition very soon)
a)

```
public static void quickSort(int[] arr, int low, int high)
{
        int pivot;
        if(high>low)
        {
                pivot = partition(arr, low, high);
                quickSort(arr, low, pivot-1);
                quickSort(arr, pivot+1, high);
        }
}
```
b)

```
public static void quickSort(int[] arr, int low, int high)
{
        int pivot;
        if(high<low)
        {
                pivot = partition(arr, low, high);
                quickSort(arr, low, pivot-1);
                quickSort(arr, pivot+1, high);
        }
}
```
c)

```
public static void quickSort(int[] arr, int low, int high)
{
        int pivot;
        if(high>low)
        {
                pivot = partition(arr, low, high);
                quickSort(arr, low, pivot);
                quickSort(arr, pivot, high);
        }
}
```
d)

```
public static void quickSort(int[] arr, int low, int high)
```

```
{
        int pivot;
        if(high>low)
        {
                pivot = partition(arr, low, high);
                quickSort(arr, low, pivot);
                quickSort(arr, pivot+2, high);
        }
}
```

View Answer

Answer: a

Explanation: Based on the pivot returned by the call to partition, recursive calls to quickSort sort the given array.

3. What is the worst case complexity of QuickSort?
a) O(nlogn)
b) O(logn)
c) O(n)
d) O(n^2)

View Answer

Answer: d

Explanation: When the input array is already sorted.

4. What is a randomized QuickSort?
a) The leftmost element is chosen as the pivot
b) The rightmost element is chosen as the pivot
c) Any element in the array is chosen as the pivot
d) A random number is generated which is used as the pivot

View Answer

Answer: c

Explanation: QuickSort is randomized by placing the input data in the randomized fashion in the array or by choosing a random element in the array as a pivot.

5. Which of the following code performs the partition operation in QuickSort?
a)

```
private static int partition(int[] arr, int low, int high)
{
        int left, right, pivot_item = arr[low];
        left = low;
        right = high;
        while(left > right)
        {
                while(arr[left] <= pivot_item)
                {
                        left++;
                }
                while(arr[right] > pivot_item)
                {
                        right--;
                }
                if(left < right)
                {
                        swap(arr, left, right);
                }
        }
        arr[low] = arr[right];
        arr[right] = pivot_item;
        return right;
}
```

b)

```
private static int partition(int[] arr, int low, int high)
{
        int left, right, pivot_item = arr[low];
        left = low;
        right = high;
        while(left <= right)
        {
                while(arr[left] <= pivot_item)
                {
                        left++;
                }
                while(arr[right] > pivot_item)
                {
                        right--;
                }
                if(left < right)
                {
                        swap(arr, left, right);
                }
        }
```

```
         arr[low] = arr[right];
         arr[right] = pivot_item;
         return right;
}
```

c)

```
private static int partition(int[] arr, int low, int high)
{
         int left, right, pivot_item = arr[low];
         left = low;
         right = high;
         while(left <= right)
         {
                  while(arr[left] > pivot_item)
                  {
                           left++;
                  }
                  while(arr[right] <= pivot_item)
                  {
                           right--;
                  }
                  if(left < right)
                  {
                           swap(arr, left, right);
                  }
         }
         arr[low] = arr[right];
         arr[right] = pivot_item;
         return right;
}
```

d)

```
private static int partition(int[] arr, int low, int high)
{
         int left, right, pivot_item = arr[low];
         left = low;
         right = high;
         while(left > right)
         {
                  while(arr[left] > pivot_item)
                  {
                          left++;
                  }
                  while(arr[right] <= pivot_item)
                  {
                          right--;
                  }
                  if(left < right)
                  {
                          swap(arr, left, right);
                  }
         }
         arr[low] = arr[right];
         arr[right] = pivot_item;
         return right;
}
```

View Answer

Answer: b

Explanation: The array is partitioned such that the elements left to the pivot are lesser than the pivot while the elements right of the pivot are greater than the pivot.

6. What is the best case complexity of QuickSort?
a) O(nlogn)
b) O(logn)
c) O(n)
d) O(n^2)

View Answer

Answer: a

Explanation: The array is partitioned into equal halves, using the Divide and Conquer master theorem, the complexity is found to be O(nlogn).

7. The given array is arr = {2,3,4,1,6}. What are the pivots that are returned as a result of subsequent partitioning?
a) 1 and 3
b) 3 and 1
c) 2 and 6
d) 6 and 2

View Answer

Answer: a

Explanation: The call to partition returns 1 and 3 as the pivot elements.

8. What is the average case complexity of QuickSort?
a) O(nlogn)

b) O(logn)
c) O(n)
d) O(n^2)
View Answer
Answer: a
Explanation: The position of partition(split) is unknown, hence all(n) possibilities are considered, the average is found by adding all and dividing by n.

9. The given array is arr = {2,6,1}. What are the pivots that are returned as a result of subsequent partitioning?
a) 1 and 6
b) 6 and 1
c) 2 and 6
d) 1
View Answer
Answer: d
Explanation: There is only one pivot with which the array will be sorted, the pivot is 1.

10. Which of the following is not true about QuickSort?
a) in-place algorithm
b) pivot position can be changed
c) adaptive sorting algorithm
d) can be implemented as a stable sort
View Answer
Answer: b
Explanation: Once a pivot is chosen, its position is finalized in the sorted array, it cannot be modified.