

Review Questions

1. linear search-simple method used for searching an array for a particular value.
binary search-Used to shorten the search time.
Interpolation search-It is same as binary search
Jump search-Since the search is performed through a jump, the time complexity is smaller than the linear search.
2. Sorting means arranging the elements of an array so that they are placed in some relevant order which may be either ascending or descending. A sorting algorithm is defined as an algorithm that puts the elements of a list in a certain order, which can be either numerical order, or any user-defined order.
3. There are Bubble sort, insertion sort, selection sort, Merge sort, Quick sort, Radix sort, Heap sort, Shell sort, Tree sort. Tree sort or bucket sort has best time complexity= $O(n)$.
4. Bubble sort algorithm is consecutive adjacent pairs of element in the array are compared with each other. Quick sort algorithms works by using a divide and conquer strategy. It selects pivot element and rearranges the element in such a way that all elements less than pivot appear before it and all elements greater than pivot appear after it. Average case of Bubble sort and Quick sort are $O(n^2)$ and $O(n \log n)$. Worst case of Bubble sort and Quick sort are $O(n^2)$ and $O(n \log n)$. Therefore, bubble sorting is more inefficient.
- 5.

(a)

77	<u>49</u>	25	12	9	33	56	81
49	77	<u>25</u>	12	9	33	56	81
25	49	77	<u>12</u>	9	33	56	81
12	25	49	77	<u>9</u>	33	56	81
9	12	25	49	77	<u>33</u>	56	81
9	12	25	33	49	77	<u>56</u>	81
9	12	25	33	49	56	77	81

(b)

11 49 25 12 9 33 56 81
9 49 25 12 11 33 56 81
9 12 25 49 11 33 56 81
9 12 25 33 11 49 56 81
9 12 25 33 49 11 56 81
9 12 25 33 49 56 11 81

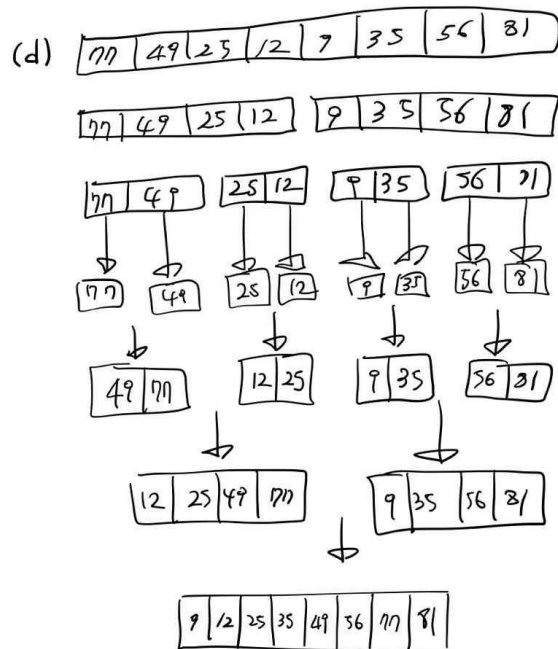
(c)

11 49 25 12 9 33 56 81
49 11 25 12 9 33 56 81
49 25 11 12 9 33 56 81
49 25 12 11 9 33 56 81
49 25 12 9 11 33 56 81
49 25 12 9 33 11 56 81
49 25 12 9 33 56 11 81

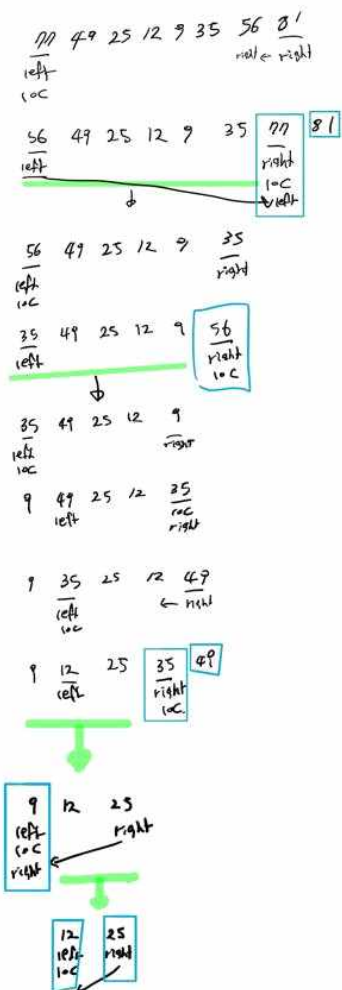
25 49 12 9 33 56 11
25 12 49 9 33 56 11
25 12 9 49 33 56 11
25 12 9 33 49 56 11

25 12 9 33 49 56
12 25 9 33 49 56
12 9 25 33 49 56

9 12 25 33 49



(e)



(f)

	0	1	2	3	4	5	6	7	8	9
17								17		
49									49	
25						25				
12			12							
9										9
35						35				
56							56			
81		81								

	0	1	2	3	4	5	6	7	8	9
17								17		
49					49					
25			25							
12		12								
9	9									
35					35					
56							56			
81									81	

(g)

17 49 25 12 9 33 49 25 12 9 8/2=4
33 56 81 17 56 81

33 49 25 12 9 25
12 9 17 → 33 49 17 (4+1)/2 = 2
56 81 56 81

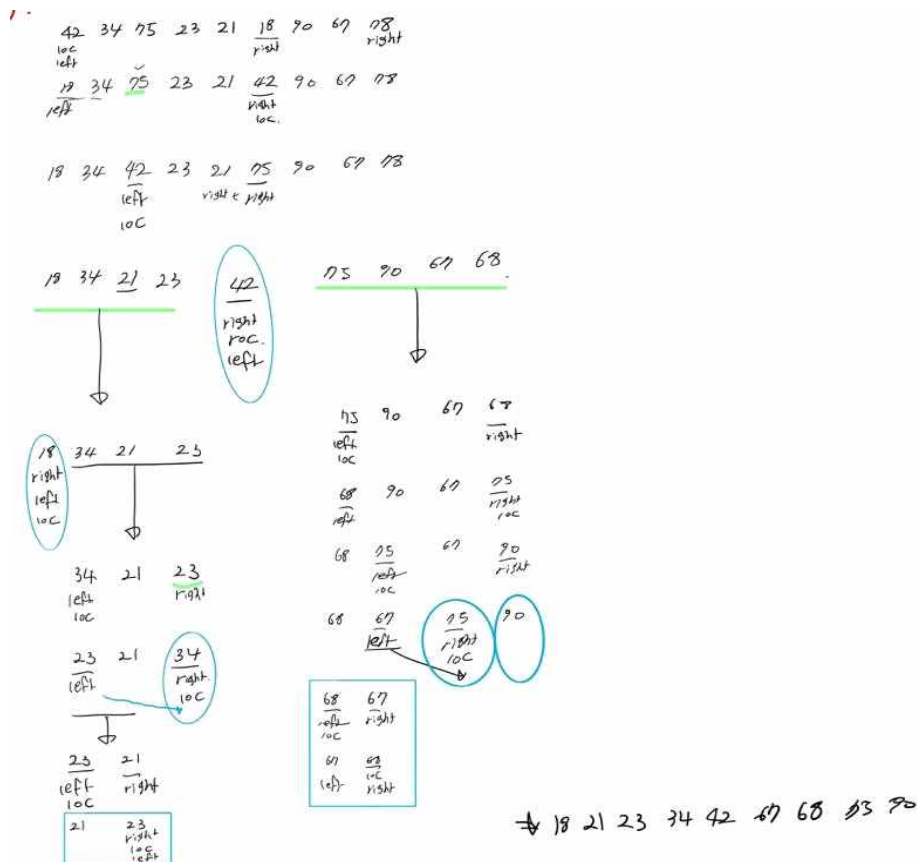
12 9 9 12 (2+1)/2 = 1
25 33 25 33
49 17 → 49 17
56 81 56 81

→ 9 12 25 33 49 17 56 81

6. Heap sort - time complexity of average case is $O(n \log n)$ and worst case is $O(n \log n)$. Quick sort - time complexity of average case is $O(n \log n)$ and worst case is $O(n^2)$.

7. The worst case occurs when the array is already sorted (ascending or descending order) and the left-most element is chosen as the pivot.

8.



9. ?

10.

1 45 27 36 54 90
1 27 45 36 54 90

selection sort

11. ?

12.

Insertion Sort ,

45 1 63 36 54 90
/ 45 63 36 54 90
/ 45 63 36 54 90

13.

```
void selection(int a[], int i, int j, int flag, int size)
{
    int temp;

    if (i < size - 1)
    {
        if (flag)
        {
            j = i + 1;
        }
        if (j < size)
        {
            if (a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
            selection(a, i, j + 1, 0, size);
        }
        selection(a, i + 1, 0, 1, size);
    }
}
```

14.

algorithm	Average case	Worst case
Bubble sort	$O(n^2)$	$O(n^2)$
Radix sort	$O(n \cdot k)$	$O(n^2 \cdot k)$
Selection sort	$O(n^2)$	$O(n^2)$
Insertion sort	$O(n^2)$	$O(n^2)$
Shell sort	-	$O(n \log n)$
Merge sort	$O(n \log n)$	$O(n \log n)$
Heap sort	$O(n \log n)$	$O(n \log n)$
Quick sort	$O(n \log n)$	$O(n^2)$

15.

- It is easy to implement and efficient to use on small sets of data
- It can be efficiently implemented on data sets that are already substantially sorted.
- It requires less memory space.
- It is said to be online, as it can sort a list as and when it receives new elements.

Multiple-choice Questions

- 1.(b)
- 2.(d)
- 3.(c)
- 4.(d)
- 5.(b)
- 6.(a)
- 7.(b)
- 8.(a)
- 9.(c)
- 10.(c)
- 11.(d)

True or False

1. T
2. T
3. T
4. F
5. T
6. T
7. F
8. F
9. T
10. T
11. F

Fill in the Blacks

1. list
2. $O(n)$
3. arranging the elements of an array
4. Bubble
5. external sorting
6. selection sort
7. $O(n^2)$
8. merge
9. $O(n \log n)$
10. $O(n \cdot k)$
11. $O(n \log n)$
12. pivot