

## Week 9: Assignment Solution

1. What is the worst case complexity of selection sort?
  - a)  $O(n \log n)$
  - b)  $O(\log n)$
  - c)  $O(n)$
  - d)  $O(n^2)$

Solution: (d)  $O(n^2)$

2. What is the best case and worst case complexity of ordered linear search?
  - a)  $O(n \log n)$ ,  $O(\log n)$
  - b)  $O(\log n)$ ,  $O(n \log n)$
  - c)  $O(n)$ ,  $O(1)$
  - d)  $O(1)$ ,  $O(n)$

Solution: (d)

Although ordered linear search is better than unordered when the element is not present in the array, the best and worst cases still remain the same, with the key element being found at first position or at last position respectively.

3. Given an array  $arr = \{12, 34, 47, 62, 85, 92, 95, 99, 105\}$  and  $key = 34$ ; what are the mid values (corresponding array elements) generated in the first and second iterations?
  - a) 85 and 12
  - b) 85 and 34
  - c) 62 and 34
  - d) 62 and 47

Solution: (b) 85 and 34

In the first iteration:  $Mid = arr[(0+8)/2] = arr[4] = 85$

In the second iteration:  $Mid = arr[(0+3)/2] = arr[1] = 34$

So, the mid values are 85 and 34

4. When the Binary search is best applied to an array?
  - a) For very large size array
  - b) When the array is sorted
  - c) When the array elements are mixed data type
  - d) When the array is unsorted

Solution: (b) Binary search is applied for sorted array.

5. Consider the array  $A[] = \{5, 4, 9, 1, 3\}$  apply the insertion sort to sort the array. Consider the cost associated with each sort is 25 rupees, what is the total cost of the insertion sort for sorting the entire array?
  - a) 25
  - b) 50
  - c) 75
  - d) 100

Solution: (c)

When the element 1 reaches the first position of the array three comparisons are only required hence  $25 * 3 = 75$  rupees.

\*step 1: 4 5 9 1 3.

## Week 9: Assignment Solution

\*step 2: 1 4 5 9 3

\*step 3: 1 3 4 5 9

6. Select the code snippet which performs unordered linear search iteratively?

- a) 

```
int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i < size; i++)
    {
        if(arr[i] == data)
        {
            index = i;
            break;
        }
    }
    return index;
}
```
- b) 

```
int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i < size; i++)
    {
        if(arr[i] == data)
        {
            break;
        }
    }
    return index;
}
```
- c) 

```
int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i <= size; i++)
    {
        if(arr[i] == data)
        {
            index = i;
            continue;
        }
    }
    return index;
}
```
- d) None of the above

Solution: (a)

Unordered term refers to the given array, that is, the elements need not be ordered. To search for an element in such an array, we need to loop through the elements until the desired element is found.

7. What will be the output?

```
#include<stdio.h>
#define func1(a,b) a > b ? b : a
#define func2(a,b); {temp=a;a=b;b=temp;}
int main()
{
    int a=3, b=5,temp;
    if((3+func1(a,b)) > b)
        func2(a,b);
    printf("%d %d", a,b);
    return 0;
}
```

- a) 3 5
- b) 3 0
- c) 5 0
- d) 5 3

Solution: (d)

Here, func1(3,5) (inside the if statement ) returns 3 as the condition given in the macro is false and it returns a=3.

Next, (3+3 > 5 ) this condition is true and the program calls func2(3,5) which swaps the values of a and b. Therefore the output is 5 3.

8. Consider an array of elements arr[5]= {5,4,3,2,1}, what are the steps of insertions done while doing insertion sort in the array.

- a) 4 5 3 2 1  
3 4 5 2 1  
2 3 4 5 1  
1 2 3 4 5
- b) 5 4 3 1 2  
5 4 1 2 3  
5 1 2 3 4  
1 2 3 4 5
- c) 4 3 2 1 5  
3 2 1 5 4  
2 1 5 4 3  
1 5 4 3 2
- d) 4 5 3 2 1  
2 3 4 5 1  
3 4 5 2 1  
1 2 3 4 5

Solution: (a)

Iteration 1: 4 is placed in its appropriate location. arr={4 5 3 2 1}

## Week 9: Assignment Solution

Iteration 2: 3 is placed in its appropriate location. arr={3 4 5 2 1}

Iteration 3: 2 is placed in its appropriate location. arr={2 3 4 5 1}

Iteration 4: 1 is placed in its appropriate location. arr={1 2 3 4 5}

9. What will be the output of the following C code?

```
#include <stdio.h>
#if A == 1
    #define B 0
#else
    #define B 1
#endif
int main()
{
    printf("%d", B);
    return 0;
}
```

- a) 0
- b) 1
- c) 01
- d) None of the above

Solution: (b)

Here, A is not initialized, so its default value is 0. The condition in the #if macro is FALSE and #define B 1 is executed. Therefore, B stores 1.

10. What will be the output?

```
#include <stdio.h>
#define a 10
int main()
{
    printf("%d ", a);
    int a=50;
    printf("%d ", a);
    return 0;
}
```

- a) 10 10
- b) 10 50
- c) 50 50
- d) Compilation error

Solution: (d)

If a is defined in macro, its value will be same throughout the program. The value assigned to that macro cannot be changed inside the program. This is the reason, it will show compilation error at the step int a=50.