Problem Solving through Programming in C
Week 09 Assignment Solution

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

Solution: (d) O(n^2)

Selection sort creates a sub-list, LHS of the 'min' element is already sorted and RHS is yet to be sorted. Starting with the first element the 'min' element moves towards the final element

2. What is the correct order of insertion sort (in ascending order) of the array arr[5]={8 3 5 9 4}?
   a) {3 8 5 9 4}→ {3 5 8 9 4}→{3 4 5 8 9}
   b) {3 8 5 9 4}→ {3 5 8 9 4}→ {3 5 8 4 9}→ {3 5 4 8 9}→{3 4 5 8 9}
   c) {3 8 5 9 4}→{3 4 8 5 9}→{3 4 5 8 9}→{3 4 5 8 9}
   d) {8 3 5 4 9}→{8 3 4 5 9}→{3 4 5 8 9}

Solution: (a) In insertion sort, we start from the second number onwards and place it in appropriate position before its current position. Thus, the correct answer is (a).

3. 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.

4. 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.

5. What is the best case and worst case complexity of ordered linear search?
    a) O(nlogn), O(logn)
    b) O(logn), O(nlogn)
    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.

6. Binary Search can be categorized into which of the following?
    a) Brute Force technique
    b) Divide and conquer
    c) Greedy algorithm
    d) Dynamic programming

Solution: (b) Divide and conquer
Since 'mid' is calculated for every iteration or recursion, we are diving the array into half and then try to solve the problem.

7. Given an array arr = {45, 77, 89, 91, 94, 98,100} and key = 100; what are the mid values (corresponding array elements) generated in the first and second iterations?
   a) 91 and 98
   b) 91 and 100
   c) 91 and 94
   d) 94 and 98

Solution: (a) 91 and 98

8. Select the appropriate pseudo code that performs selection sort

a) int min;
```
    for(int j=0; j<arr.length-1; j++)
    {
            min = j;
            for(int k=j+1; k<=arr.length-1; k++)
            {
                    if(arr[k] < arr[min])
                            min = k;
            }
            int temp = arr[min];
            arr[min] = arr[j];
            arr[j] = temp;
}
```

b) int min;
```
    for(int j=0; j<arr.length-1; j++)
    {
            min = j;
            for(int k=j+1; k<=arr.length; k++)
            {
                    if(arr[k] < arr[min])
                            min = k;
            }
            int temp = arr[min];
            arr[min] = arr[j];
            arr[j] = temp;
    }
```

c) int min;
```
    for(int j=0; j<arr.length-1; j++)
    {
            min = j;
            for(int k=j+1; k<=arr.length-1; k++)
            {
                    if(arr[k] > arr[min])
                            min = k;
            }
            int temp = arr[min];
            arr[min] = arr[j];
            arr[j] = temp;
    }
```

d)
```
int min;
for(int j=0; j<arr.length-1; j++)
{
        min = j;
        for(int k=j+1; k<=arr.length; k++)
        {
                if(arr[k] > arr[min])
                        min = k;
        }
        int temp = arr[min];
        arr[min] = arr[j];
        arr[j] = temp;
}
```

Solution: (a)
Starting with the first element as ‘min’ element, selection sort loops through the list to select the least element which is then swapped with the ‘min’ element.

9. Consider the array A[]= {5,4,9,1,3} apply the insertion sort to sort the array . Consider the cost associated with each sort (swap operation) is 25 rupees, what is the total cost of the insertion sort when element 1 reaches the first position of the array?

   a) 25
   b) 50
   c) 75
   d) 100

   Solution: (b)

   When the element 1 reaches the first position of the array two comparisons are only required hence 25 * 2= 50 rupees.

   *step 1: 4 5 9 1 3.

   *step 2: 1 4 5 9 3

10. Find the output of the following C program
```
#include<stdio.h>
int main()
{
    int a;
    int arr[5] = {1, 2, 3, 4, 5};
    arr[1] = ++arr[1];
    a = arr[1]++;
    arr[1] = arr[a++];
    printf("%d,%d", a, arr[1]);
    return 0;
}
```

   a) 5,4
   b) 5,5
   c) 4,4
   d) 3,4

Solution: (c)