Assignment Solutions | Bubble sorting | Week 9

1. Which of the following(s) is/are true about bubble sort:
    1. It is stable sort
    2. It has a worst case space complexity of O(n)
    3. It involves swapping of adjacent elements
    4. After each iteration, the greatest element is placed at the end of the array.

Solution :

```
1st , 3rd and 4th are correct options.
```

2. What will the following array look like after one iteration of bubble sort [1,6,2,5,4,3].
    1. [1,3,2,4,5,6]
    2. [1,2,3,4,5,6]
    3. [1,2,5,4,3,6]
    4. [1,2,4,5,3,6]

Solution :

```
3rd option is correct.
```

3. In which case does bubble sort works in the most efficient way:
    1. When the array is sorted in increasing order
    2. When the array is sorted partially
    3. When the array is sorted in decreasing order.
    4. When the array is nearly sorted.

Solution :

```
1st option is correct.
```

4. Sort the array in descending order using Bubble Sort.

Solution :

```
#include <iostream>
using namespace std;
int main() {
    int arr[5]={7,2,32,5,20};
    int size=5;
    for (int i = 0; i < size - 1; ++i){
      for (int j = 0; j < size - i - 1; ++j){
        if (arr[j] < arr[j + 1]){
          int temp = arr[j];
          arr[j] = arr[j + 1];
          arr[j + 1] = temp;
        }
      }
    }
    for (int i = 0; i < size; ++i){
      cout<<arr[i]<<" ";
    }
return 0;
}
```

5. Check if the given array is almost sorted. (elements are at-most one position away).

Solution :

```
#include <iostream>
using namespace std;
int main() {
    int A[5]={7,2,32,5,20};
    int n = 5;

    for (int i = 0; i < n - 1; i++) {
        if (A[i] > A[i + 1]) {
            swap(A[i], A[i + 1]);
            i++;
        }
     }
    int i;
    for (i = 0; i < n - 1; i++)
      if (A[i] > A[i + 1]) {
        cout<<"No"<<endl;
        break;
      }
    if(i == n - 1)cout<<"Yes"<<endl;
    return 0;
}
```
