Algorithm

What is the Big-O Time Complexity Analysis of BubbleSort?

```
class BubbleSort
{
    void bubbleSort(int arr[])
    {
        int n = arr.length;
        for (int i = 0; i < n-1; i++)
            for (int j = 0; j < n-i-1; j++)
                if (arr[j] > arr[j+1])
                {
                       // swap arr[j+1] and arr[i]
                       int temp = arr[j];
                       arr[j] = arr[j+1];
                       arr[j+1] = temp;
                   }
       }

       /* Prints the array */
       void printArray(int arr[])
       {
           int n = arr.length;
           for (int i=0; i<n; ++i)
               System.out.print(arr[i] + " ");
           System.out.println();
       }

       // Driver method to test above
       public static void main(String args[])
       {
           BubbleSort ob = new BubbleSort();
           int arr[] = {64, 34, 25, 12, 22, 11, 90};
           ob.bubbleSort(arr);
           System.out.println("Sorted array");
           ob.printArray(arr);
       }
   }
```

Description=> This is one way to do sorting using Bubble sort. We are using two nested loops here. Inner loop elements are compared and swapped and number of operations occurred in inner loop are n. As well as number of iterations happened in outer loop are n. So now this function has quadratic running time of O(n^2) .