ALGORITHM AND GRAPH FOR INSERTION SORT:

```
class InsertionSort {
	/*Function to sort array using insertion sort*/
	void sort(int arr[])
	{
		int n = arr.length;
		for (int i = 1; i < n; ++i) {

			int key = arr[i];
			int j = i - 1;

			/* Move elements of arr[0..i-1], that are
			greater than key, to one position ahead
			of their current position */
			while (j >= 0 && arr[j] > key) {

				arr[j + 1] = arr[j];
				j = j - 1;
			}
			arr[j + 1] = key;
		}
	}

	/* A utility function to print array of size n*/
	static 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
	public static void main(String args[])
	{
		int arr[] = { 12, 11, 13, 5, 6 };

		InsertionSort ob = new InsertionSort();
		ob.sort(arr);

		printArray(arr);
	}
}
```

X	Y
10	969.755546915104
500	110150.732861087
1000	345226.065084094
1500	705216.722254921
2000	1190122.70437357