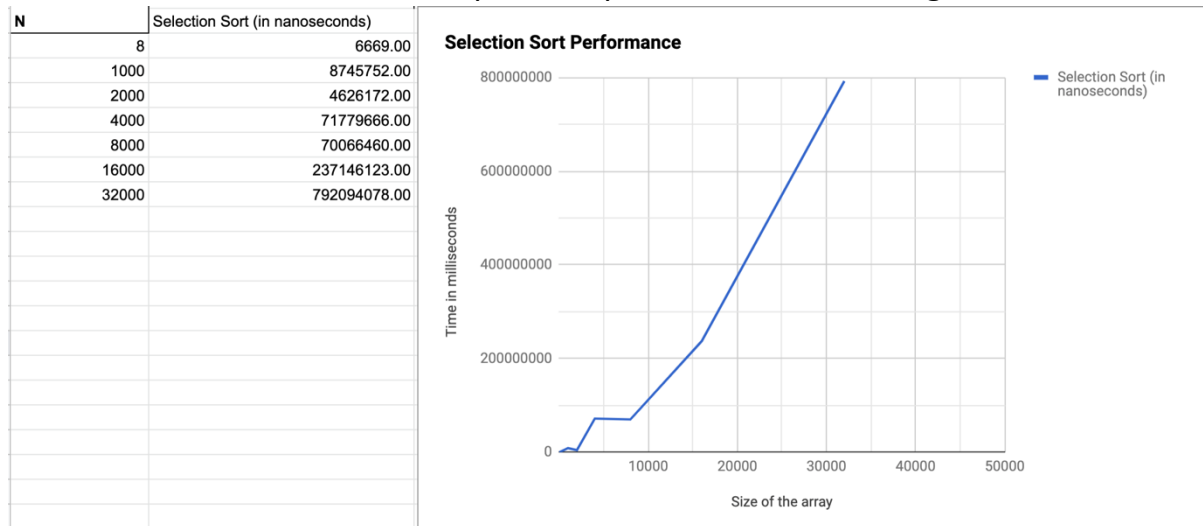


## Selection Sort

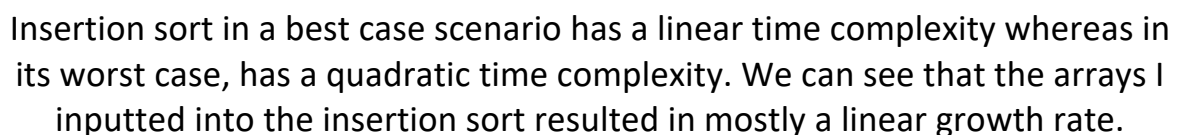
```
public void sort(int[] arr){
    int n = arr.length;
    for (int i = 0; i < n-1; i++){
        int minNum = i;
        for (int j = i+1; j < n; j++){
            if (arr[j] < arr[minNum]) minNum = j;
        }
        int temp = arr[minNum];
        arr[minNum] = arr[i];
        arr[i] = temp;
    }
}
```

Table and Graphical representation of timings



A selection sort algorithm has a quadratic time complexity  $O(n^2)$ . The graph displays a quadratic time growth once the inputted array size reaches past 8000.

Insertion sort works by comparing the current place (i) in the array with the value prior to it (i-1). If the value at i-1 is less than then the value at i then swap these two values within the while loop. The while loop doesn't break if it finds that the value prior to j is also less than the value at j-1 meaning these two values will also be swapped.

[illegible]

## Stalin Sort

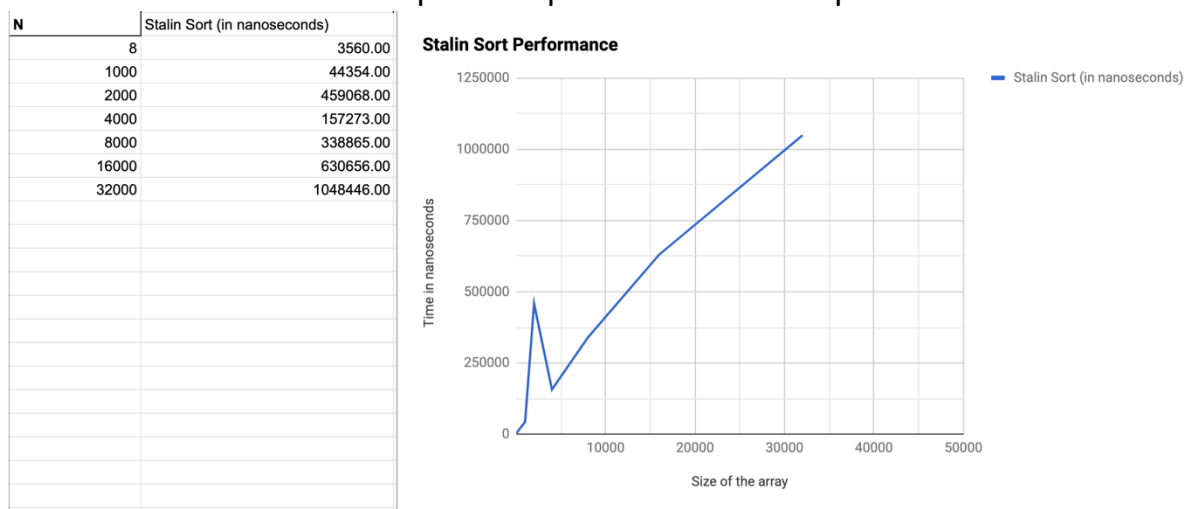
```
public int[] removeOutOfOrderElements(int[] arr, int size){
    int[] temp = new int[size];
    int j = 1;
    temp[0] = arr[0];

    for (int i = 1; i < size; i++){
        if (temp[j - 1] <= arr[i]){
            temp[j] = arr[i];
            j++;
        }
    }

    return temp;
}
```

Stalin sort is a silly type of sorting algorithm that simply removes every value that's out of the order.

Table and Graphical representation of acquired times



This shows that a Stalin sort algorithm has a linear time complexity  $O(n)$ .