```
100 Classes
profileSelection::allocs:3, compares:4950, swaps:99
profileBubble::allocs:3, compares:4950, swaps:2343
profileInsertion::allocs:3, compares:2343, swaps:2343
profileQuicksort::allocs:809, compares:783, swaps:286

1000 Classes
profileSelection::allocs:3, compares:499500, swaps:999
profileBubble::allocs:3, compares:499500, swaps:248567
profileInsertion::allocs:3, compares:248567, swaps:248567
profileQuicksort::allocs:8225, compares:13984, swaps:7827

10000 Classes
profileSelection::allocs:3, compares:49995000, swaps:9999
profileBubble::allocs:3, compares:49995000, swaps:24929215
profileInsertion::allocs:3, compares:24929215, swaps:24929215
profileQuicksort::allocs:105161, compares:201583, swaps:75984
```

Selection sort works by going through the entire array, finding the smallest element, and then swapping that into the position of the first unsorted element. By repeatedly finding the smallest element in the shrinking subset, and then swapping it with the first unsorted element, the list becomes ordered from least to greatest. The amount of times an element is compared to another element is significantly greater than the amount of swaps needed to be done, since the code is running through the entire array, comparing every unsorted element with the selected “smallest” element. This is because the code only swaps once per outer loop iteration, but compares in every