

## Run time analysis

### Part 1: Analyze code and guess the run time

For my original code, I set my index 0 as my minimum. Then the rest of the elements in the list need to keep comparing with index 0. The total run times is about  $O(n^2)$ . The reason why I make this assumption is that I have nested loop and each element in the list compares with the minimum one.

Input Sizes (list size)	Predicted Time	Actual Run Time	Predict Time (sec)
n = 2500	6250000	4.12 S	5 S
n = 3500	12250000	4.42 S	7 S
n = 4500	20250000	4.77 S	9 S

For my updates code, it compares with previous element. If it smaller than previous element, the index of the element will become the previous element. It will save significantly running time because the previous elements already sorted. The best will be n time, the worst about  $n^2$ . Average running time:  $(n+n^2)/2$ .

Input Sizes (list size)	Best Scenario	Worst Scenario	Average Run times	Actual Run Time	Predict Time (sec)
n = 2500	2500	6250000	3125000	1.1 S	3 S
n = 3500	3500	12250000	6125000	1.2 S	4 S
n = 4500	4500	20250000	10125000	1.22 S	5 S

In conclusion, my updated code's running time is about 50% previous running time.