5. Sorting Algorithms :: Selection Sort :: Time Complexity

How efficient is selection sort? Let's take an example. Suppose we are sorting $list = \{1, 2, 3, 4, 5\}$ in ascending order:

startIndex	***findMinIndex*()** loops**	**swaps**
0	4	1
1	3	1
2	2	1
3	1	1

The key operation that requires the most time is to find the index of the minimum element, i.e., it is this operation:

```
pList.get(i).compareTo(pList.get(minIndex)) < 0
```

For our 5-element list the number of comparisons is $1 + 2 + 3 + 4 = 10$. For an n-element list the number of comparisons would be:

$$c(n) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$$

which is the same function we derived in analyzing insertion sort, thus proving that selection sort also has worst case time complexity $O(n^2)$.