Selection Sort:

General Description: Selection Sort is probably the most immediately intuitive sorting method and involves iterating through the array and finding the new minimum of the unsorted section of the array. At the beginning, all the elements are part of the unsorted section. With each pass, the sorted section of the array is incremented. We scan the unsorted section of the array for a new minimum during each pass. At the end of the pass, we have the minimum of the unsorted section, and swap it with the element at the beginning of the unsorted section. Then the sorted section grows by one element. Conversely, the unsorted section shrinks by one element. This is kept track of with the outer for loop. Like bubble sort, there are two for loops, one nested within another. Therefore, the time complexity is O(n^2). Space complexity is O(1) since we don't have to keep track of any auxiliary arrays. It is an in-place sorting algorithm that does not require extra space in memory.

Selection Sort Code:

```
for i in range(len(A)): #outer loop keeps track of the passes

    # Find minimum element in the
    # unsorted section of the array
    min_idx = i
    for j in range(i+1, len(A)): # inner loop compares and finds min
        if A[min_idx] > A[j]:
            min_idx = j

    # At the end of the pass, swap the minimum element with
    # the first element of the unsorted section of the array
    A[i], A[min_idx] = A[min_idx], A[i]
```