Correctness of Selection Sort

Selection Sort is an algorithm to sort numbers in an array. It works by going through the array multiple times and picking the smallest number in the array, then putting it where it fits. This process is repeated until everything is sorted.

How Selection Sort Works

1. Begin the process:
   - First, the array is unsorted.
   - In every step of the code, the algorithm finds the smallest number from the array which is unsorted and moves it to the front of that array. Over time, the sorted part of the array grows by every step.
2. Loop Invariant:
   - A loop invariant is just something that stays true while the algorithm is running.
   - For Selection Sort, it means that at any step:
     - The sorted part of the array remains in correct order.
     - The unsorted part of the array contains numbers that have not been placed yet.
   - As the algorithm continues, the sorted part of the array keeps getting bigger while the unsorted part gets smaller.

Why It Works

- Step-by-Step Process:
  - First Step: The smallest number in the entire array is found and swapped with the number at index 0. This means the number at index 0 is in correct position.
  - Second Step: Now, the smallest number in the remaining part of the array is found and swapped with the number of index 1.
  - More Steps: This process keeps going. Every time, the smallest number from the unsorted part of the array is placed in its correct spot.
  - End of Sorting: When there is only one number left in the unsorted part of the array, it is automatically in the correct place. This is when the sorting is done.

Example

Let's take the array: [27, 1, 25, 7, 3, 23].

- First Round: Find the smallest number (1) and swap it with the number at index 0.
  Array becomes: [1, 27, 25, 7, 3, 23].
- Second Round: From the unsorted part of the array [27, 25, 7, 3, 23], the smallest number is 3. Swap it with 27.
  Array becomes: [1, 3, 25, 7, 27, 23].
- Third Round: From the unsorted part of the array [25, 7, 27, 23], the smallest number is 7. Swap it with 25.
  Array becomes: [1, 3, 7, 25, 27, 23].
- Fourth Round: From the unsorted part of the array [25, 27, 23], the smallest number is 23. Swap it with 25.
  Array becomes: [1, 3, 7, 23, 27, 25].
- Fifth Round: From [27, 25], the smallest number is 25. Swap it with 27.
  Array becomes: [1, 3, 7, 23, 25, 27].
- Last Step: The final number 27 is already in the correct position.
  The sorted array is now [1, 3, 7, 23, 25, 27].

Final Thoughts

Selection Sort is a really simple sorting algorithm that always works by building the left array into the sorted part of the array. It may not be the fastest for the large arrays, but it is very easy to use and understand. By the end of the algorithm, every number ends up in the correct position, resulting in a fully sorted array.