2. Argue selection sort correctness.

Algorithm:

```
for i = 1 to A.length - 1
    min = i
    for j= i to A.length - 1
        if A[j] < A[min]
            min = j
        endif
    endfor
   swap (A[i],A[min])
endfor
```

loop invariant:

At the start of each iteration for the outer for loop,
A[1,2,….,i-1].A[i].A[i+1,i+2,….,n] where n is the length of the array then we have the array A[1 . . . i-1] consists of the sorted elements also which are the smallest and the array A[i,..n] contains the remaining elements of A.

Initialization:

At the first iteration, i = 1, the array to the left of i is empty as there is only one element. So it can be considered that an empty array is ordered and obeys the loop invariant.
A[1]—only one element hence is already sorted

Maintenance :

For the next iterations, all the elements element from 1 to i-1 are already sorted and will not be taken into consideration. For the remaining array, the inner for loop will find the smallest element in A[i . . . n] and will swap it with the A[i]th element.
This element is swapped into the i th position. The new A[i] element will ne grater than any of the element in A[1 . . . i − 1] as the loop invariant is true prior to the start of the iteration, so it will simultaneously be the smallest element from i to the right and bigger than any element to its left. So The loop invariant is true.

Termination:

A[1,2,….n-1].A[n]
For the last iteration as we know the array A[1,…n] is sorted with the elements smaller than the right side of the array which is A[n]. so there will be no need to swap and the array is already sorted and hence the loop invariant is true

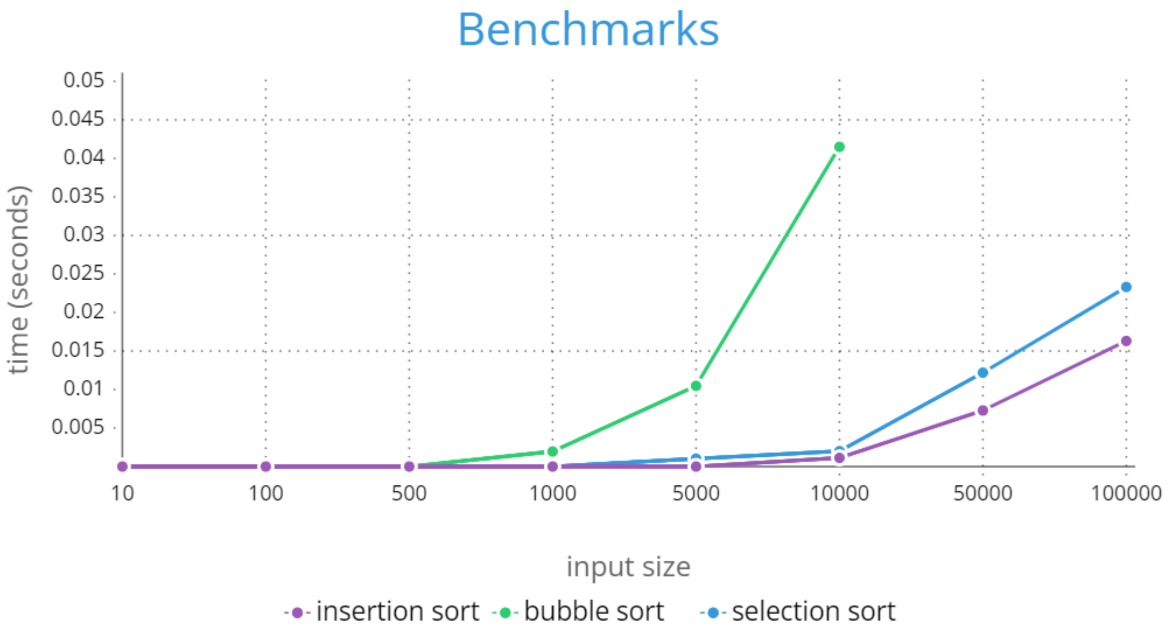
System information:

Processor	12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz
Installed RAM	16.0 GB (15.7 GB usable)
System type	Windows 11 64-bit operating system, x64-based processor
Python version	3.12.1

Benchmarks:

On running the sorting algorithms with 7 inputs each of sizes 10,100,1000,5000,10000,50000,100000 we get the following values (Very small values are rounded off to 0.0)

	10	100	1000	5000	10000	50000	100000
Selection	0.0	0.0	0.0	0.001024723	0.0020058155	0.0121893882	0.0233108997
Bubble	0.0	0.0	0.0019524097	0.0104765892	0.041500568	1.051724195	4.6182520389
Insertion	0.0	0.0	0.0	0.0	0.0011231899	0.0072762966	0.0163037776


Benchmarks
time (seconds)
input size
insertion sort
bubble sort
selection sort