



While sorting an increasing data,

For insertion sort: when the program loops through every element of the data, it will always check if the former number is greater than the current one. If so, it will continue checking the former ones until no former values are greater. However, because the data is already in an increasing order, the checking procedure basically just ends at the first time. And all the supposedly following execution (assigning new values to the former elements) will not be executed. Therefore, in this case, insertion sort basically just loops through the data and do very few executions.

For selection sort: the program starts with looping through every element of the data as well, but every time it goes through a single element, it starts another loop which goes through every element after it and find the minimum value among them.

Unlike insertion sort, no matter if the data is pre-sorted or not, selection sort will always go with a second loop inside the first loop until the first loop is ended. Therefore, for a initially increasing data, insertion sort always take fewer steps than selection sort.

Because in such case insertion sort can be treated as only looping the data once, the number of instructions increase linearly with the size of the data, so the line is a straight line (not flat). In contrary, the instructions of selection sort increase quadratically because it gets into another loop while iterating every element, which can be deemed as a loop inside a loop.