**Step 1) In your own words, describe what sorting is in general.**

Sorting is an algorithm that puts elements in a list into an order. The most used orders are ascending and descending orders. Sorting a list allows for easier analysis and visualisation. If I wanted to search for a value in a list, sorting the data makes it very efficient using binary search.

**Step 2) Research sorting algorithms. Describe advantages and disadvantages for at least three different sorting algorithms. Please provide references for external resources.**

Bubble sort

* It is popular and easy to implement.
* Elements of the list are swapped in place without using additional temporary storage.
* Bubble sort is not suitable for most real-life applications.
* Not suitable for huge data sets because it works by repeatedly stepping through the list to be sorted and it compares two items at a time.

Quicksort

* Quick sort is regarded as the best sorting algorithm because it works well with a huge list of items.
* It sorts in place without requiring additional storage.
* If the list were sorted, then bubble sort would be more efficient than quick sort.
* If the items in the list were integers, then the radix sort is more efficient.

Selection sort

* It works well on a small list.
* It is an in-place sorting algorithm that does not require additional temporary storage.
* The efficiency is very poor when sorting a big list of items.
* Quick sort is much more efficient than selection sort.

*“Sorting Algorithm”, Programiz,*

<https://www.programiz.com/dsa/sorting-algorithm>

*“Searching and Sorting”*, Aissmspoly,

<https://aissmspoly.org.in/wp-content/uploads/2020/01/Searching-and-Sorting.pdf>

**Step 3) In your own words, describe why you generally need comparison operators to successfully sort a list of objects. In addition, describe how you could sort a list of objects without adding comparison operators.**

Generally, we need to use comparison operators to sort a list of items. Comparison operators establish the relative ordering between the elements. Sorting is usually in ascending or descending order, and by comparing the previous item and the next item in the list, we can re-order the items in an order.

We could sort a list of objects without using comparison operators. First, we give ‘value’ attribute to an object class. Create a function that returns the value of the object. Then create a function that sorts based on the value of the object using counting sort (which is a non-comparative sorting algorithm). Once the sorting of the values is done, reconstruct the list by placing the objects based on their sorted values.

**Step 4d) Choose an algorithm of your liking based on the answers above and describe why you chose it.**

All the sorting algorithms will do the job, and it is not included in the answers above, but I chose the Radix sorting algorithm because while the size of the list can vary, the elements of the list will only be positive integers, so Radix sort would be an efficient choice.