Advanced Programming

Assignment 3

Benjamin Cooper

**Step 1 – Knowledge Question (40-70 words) In your own words, describe what sorting is in general.**

An algorithmic process for putting elements into an order, typically sorted by ascending or descending values.

**Step 2 – Knowledge Question (60-100 words) Research sorting algorithms. Describe advantages and disadvantages for at least three different sorting algorithms. Please provide references for external resources.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **Algorithm** | **ADVANTAGE** | **DISADVANTAGE** | **REFERENCE** |
| **MERGE SORT** | 1. Divide list into sub lists until lists of size 1(single item) 2. Merge sub lists upwards into ordered lists 3. Each upwards merge merges the ordered lists together into a larger ordered list until all items sorted | **Relatively fast and stable** | **Requires additional memory for sub lists** | [**StudySmarter UK**](https://www.studysmarter.co.uk/explanations/computer-science/algorithms-in-computer-science/merge-sort/) |
| **BUBBLE SORT** | 1. **Start with item at position x=0, if larger than item at position x+1, switch, if smaller, leave in same order** 2. **Iterate over the entire list, so that the item initially at x=0 is in a position higher than all elements smaller than it.** 3. **Repeat for x=1, x=2… until whole list has been sorted** | **Easy to implement, low memeoyr required** | **Very slow sorting time relative to other sorting algorithms** | [**GeeksforGeeks - Bubble sort**](https://www.geeksforgeeks.org/bubble-sort/) |
| **QUICK SORT** | 1. **Pivot element selected in list** 2. **All elements larger and smaller go into two separate lists respectively** 3. **These sub lists are then partitioned again into 2 sub lists either larger or smaller than the pivot.** 4. **Sub lists then merged back together with the pivot value between them** | **Efficient for large datasets, low memeory required** | **Inefffieicnt for smaller datasets** | [**GeeksforGeeks - Quick Sort**](https://www.geeksforgeeks.org/quick-sort/) |

**Step 3 – Knowledge Question (20-50 words) 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**

Comparison operators are required to sort a list to ensure that the elements are consistently larger or smaller than their adjacent elements

Sorting without comparison operators may exist where we are not sorting by relative value.

e.g. sorting into “buckets” based on a shared attribute such as colour or location.

**4.d Implement a static method for the Player class that will sort a list of Player objects in descending order (higher scores come first). Choose an algorithm of your liking based on the answers you provided to the Knowledge Questions and describe why you chose it. IMPORTANT: you must implement this algorithm yourself; you may not use Python’s sort method**

I have implemented a function called scoresort in a file called scoresort.py, this function will sort an array of objects that have an object.score attribute. By default in ascending order, but setting AscendingOrder to be 0, the function will sort in reverse.

This function is defined in a separate file for readability’s sake. It is then imported and called in the player file.

Scoresort is an implementation of the quicksort algorithm.

I chose to implement quicksort because I thought it was the most sophisticated algorithm and therefore presented the best learning opportunity.

it is a great algorithm in terms of memoery and speed for large datasets, but in the case of a smaller dataset, it will not be the fastest when compared to other sorting algorithms