**Sorting Algorithms**

**Handout 1 – Sorting Algorithms**

**Task 1 - Bubble sort**

1. For each of the below, show the stages of the bubble sort when applied to the data sets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chris | Ben | Fran | Vishwa | Neil | Lauran |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pass 1 |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pass 2 |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. How many passes would it take to put the following list in ascending alphabetical order?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 1 | 8 | 2 | 3 |

Number of passes = 3 (2 to sort and one for final check)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass 1 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass 2 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. For each of the below, show the stages of the bubble sort when applied to the data sets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 7 | 2 | 6 | 2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass 1 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass 2 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass 3 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Task 2 - Merge sort**

1. Show the steps that a Merge sort would take to put the following names into ascending alphabetical order (from A to Z).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chris | Ben | Fran | Vishwa | Neil | Lauran |

|  |
| --- |
|  |

1. Show the steps that a Merge sort would take to put the following numbers into ascending alphabetical order (e.g from 1 to 100).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 1 | 4 | 3 | 5 | 6 | 90 | 55 | 23 | 27 |

|  |
| --- |
|  |

**Task 3 - Insertion sort**

1. Show the steps that an Insertion Sort would take to put the following list into ascending alphabetical order (from A to Z).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **sorted** | **unsorted** | | | | | |  |
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| value | Sydney | Salt Lake City | Athens | Turin | Beijing | Vancouver | London |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. Show the steps that an Insertion Sort would take to put the following list into ascending numerical alphabetical order (e.g. from 1 to 10).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **sorted** | **unsorted** | | | | |
| index | 0 | 1 | 2 | 3 | 4 |
| value | 15,15 | 25,25 | 13,13 | 29,29 | 18,18 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. An insertion sort is used to put the following words into ascending alphabetical order (from A to Z).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pumpkin | flour | wall | house | wall |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Tick (✓) **one** box in each row to identify whether each statement about the insertion sort is true or false.

|  |  |  |
| --- | --- | --- |
| **Statement** | **True (✓)** | **False (✓)** |
| The list of words is initially split into a sorted portion and an unsorted portion. |  |  |
| The insertion sort uses a divide stage and then a conquer stage. |  |  |
| The list of words must be in order before the insertion sort can start. |  |  |
| Each word is inserted into the correct place in the array, one by one. |  |  |
| The insertion sort will not work because the word “wall” appears twice. |  |  |

**Task 4 – Comparison**

|  |  |  |
| --- | --- | --- |
| **Insertion Sort**  [**http://tiny.cc/insert-ani**](http://tiny.cc/insert-ani) | **Bubble Sort**  [**http://tiny.cc/bubble-ani**](http://tiny.cc/bubble-ani) | **Merge Sort**  [**http://tiny.cc/merge-ani**](http://tiny.cc/merge-ani) |
|  |  |  |

**Task 5 – Gameboard**

Complete the assignment (gameboard) on the Isaac Computer Science website.

<https://isaaccs.org/assignment/9c661cf8-3cbb-41b8-a7a9-7e604041c77c>

**Further Reading Isaac Computer Science**

For further reading on sorting algorithms visit the Isaac Computer Science website-

<https://isaaccomputerscience.org/topics/sorting?examBoard=all&stage=all>