**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 |

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

Pass 1

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

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)

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

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

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

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

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

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

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

**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 |

|  |
| --- |
| Split the items into individual items. (Make sure to show in an exam all steps)  Rebuild list putting items in order.  Ben, Chris Fran, Vishwa Lauran, Neil  Ben, Chris, Fran, Vishwa Lauran, Neil  Ben, Chris, Fran, Lauran, Neil, Vishwa |

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 |

|  |
| --- |
| Split the items into individual items.  Rebuild list putting items in order.  100 1 4 3 5 6 90 55 23 27  1, 100 3, 4 5, 6 55, 90 23, 27  1, 3, 4, 100 5, 6, 55, 90 23, 27  1, 3, 4, 6, 55, 90, 100 23, 27  Final ordered list = 1, 3, 4, 6, 23, 27, 55, 90, 100 |

**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 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sydney |  |  |  |  |  |  |
| Salt Lake City | Sydney |  |  |  |  |  |
| Athens | Salt Lake City | Sydney |  |  |  |  |
| Athens | Salt Lake City | Sydney | Turin |  |  |  |
| Athens | Beijing | Salt Lake City | Sydney | Turin |  |  |
| Athens | Beijing | Salt Lake City | Sydney | Turin | Vancouver |  |
| Athens | Beijing | London | Salt Lake City | Sydney | Turin | Vancouver |

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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15,15 | 25,25 |  |  |  |
| 13,13 | 15,15 | 25,25 |  |  |
| 13,13 | 15,15 | 25,25 | 29,29 |  |
| 13,13 | 15,15 | 18,18 | 25,25 | 29,29 |

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pumpkin |  |  |  |  |
| flour | pumpkin |  |  |  |
| flour | pumpkin | wall |  |  |
| flour | house | pumpkin | wall |  |
| flour | house | pumpkin | wall | 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. | X |  |
| The insertion sort uses a divide stage and then a conquer stage. |  | X |
| The list of words must be in order before the insertion sort can start. |  | X |
| Each word is inserted into the correct place in the array/list, one by one. | X |  |
| The insertion sort will not work because the word “wall” appears twice. |  | X |

**Task 4 – Comparison**

A close-up of a bubble sort

Description automatically generated

**Task 5 – Gameboard**

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

​

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