

BubbleSort: Starting from the left, compare adjacent elements, swap to make bigger one to the right, next pass is (n – 1), ShortBubbleSort: if no swaps were made in a pass, then it is sorted.

SelectionSort: In each pass, look for the biggest element and place it at the end, next pass is (n – 1).

InsertionSort: In each pass, take the next element and place it in the right position in the sorted portion.

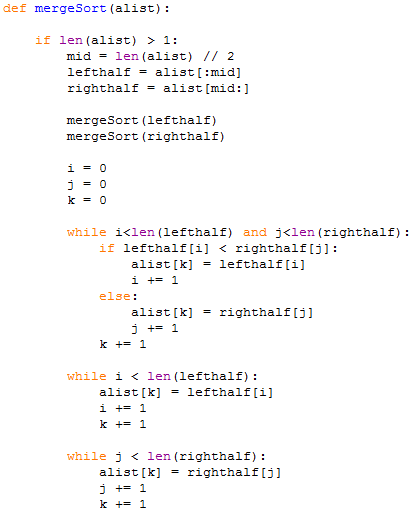
ShellSort: Break the list into sections (k), and do insertion sort with that gap.

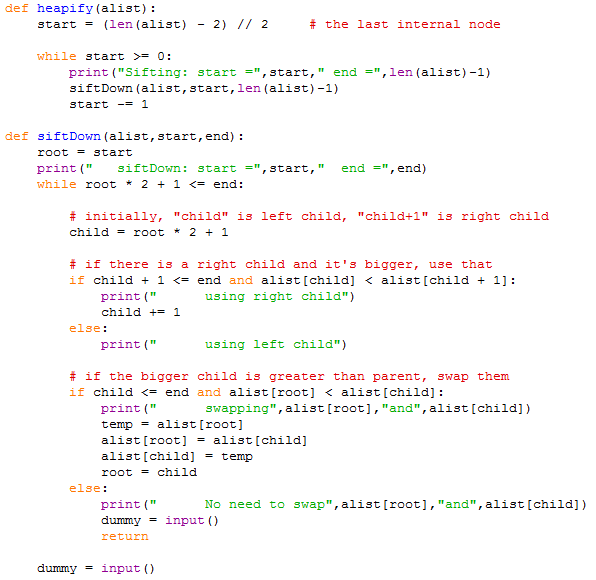
MergeSort: Break list into half until only one element, and merge with recursion.

QuickSort: Pick pivotpoint, and have a leftmark and rightmark, move them until the rightmark is on the left of leftmark, and then swap pp with the rightmark, then make recursive call on left side, and then on right.

HeapSort: Make a max heap, and then swap root with the last leaf node and remake a heap by calling siftDown. Heapify is O(n) but calls siftDown too, so it’s O(n log n), and in phase 2, siftDown is O(logn) and is called n times, so overall it’s O(n log n).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best Case | Worst Case | Avg Case |
| Sequential Search | item is first: O(1) | item is last/not there: O(n) | O(n) |
| Binary Search | item is in center: O(1) | item is not there: O(log n)/O(n log n) with recursion | O(log n) |





Source: a node where you can’t get to but can come from (arrows pointing from it).

Sink: a node where you can get to but can’t get from (arrows pointing to it).

Adjacency Matrix: empty column is source, empty row is sink. The x-axis are the ending vertices, y-axis are the beginning vertices.

PreOrder: root, leftChild, rightChild

InOrder: leftChild, root, rightChild

PostOrder: leftChild, rightChild, root

BinarySearchTree: Average: access, search, insert, deletion are all O(log n), Worst: all O(n).

AVL Tree: Average and Worst: access, search, insert, deletion are all O(log n).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best Case | Worst Case | Avg Case |
| BubbleSort | sorted: O(n) | reverse sorted: O(n^2) | O(n^2) |
| ShortBubbleSort | sorted: O(n) | reverse sorted: O(n^2) | O(n^2) but can stop early |
| SelectionSort | no best/worst case: O(n^2) | sorted: O(n^2) | O(n^2) |
| InsertionSort | sorted: O(n) | reverse sorted: O(n^2) | O(n^2) |
| ShellSort | O(n log n) | O(n(log n)^2) | O(n(log n)^2) |
| MergeSort | O(n log n) | O(n log n) | O(n log n) |
| QuickSort | O(n log n) | O(n log n) | sorted/reversed: O(n^2) |
| HeapSort | O(n log n) | O(n log n) | O(n log n) |

