

cycle_7_lab_exercises

February 27, 2021

1 Cycle 7 Lab Exercises - Sorting

This week, you will practice tracing BubbleSort, implement an iterative version of selection sort, and then implement a sorting algorithm you probably have not seen before.

1.0.1 Task 1 - Understanding BubbleSort:

Look back at the BubbleSort algorithm we looked at in class, and trace its execution when called with: `bubbleSort([2, 1, 3, 5, 1, 7])` Show each step. How many comparisons does it perform? Add a line to `bubbleSort` to count how many swaps it performs, and then try it out on a variety of lists. What is worst - a sorted list, a random list, or a reverse-sorted list?

1.0.2 Task 2 - Implementing Iterative SelectionSort:

We talked briefly about a recursive SelectionSort algorithm in lecture. I would like you to implement an iterative version of SelectionSort. Feel free to look at the recursive version we went over, but try to avoid looking up an iterative version and copying it. It may help you to write out the steps that Selection Sort should take in its execution and then try to transfer those to code.

1.0.3 Task 3 - Implementing a new sorting algorithm - the odd-even sort

The odd-even sort algorithm is based on repeatedly making two passes on a list: 1. The first pass examines pairs of items $a[j]$, $a[j+1]$ where j is odd. 2. The second pass examines pairs of items $a[i]$, $a[i+1]$ where i is even. 3. In each pass, the following occurs: if the contents of the pair being examined are unordered, then the elements are swapped. 4. Repeat passes until the list is sorted.

Consider the following list: `[1,7,5,2,6]` First pass * Start with odd position * Examine $a[1]$ and $a[2]$, compare contents of this pair (7 and 5): swap elements. Resulting list: `[1,5,7,2,6]` * Examine $a[3]$ and $a[4]$, compare contents of this pair (2 and 6): do not swap elements. * Start with even position * Examine $a[0]$ and $a[1]$, compare contents of this pair (1 and 5): do not swap elements. * Examine $a[2]$ and $a[3]$, compare contents of this pair (7 and 2): swap elements. Resulting list: `[1,5,2,7,6]` Repeat until list is sorted.

[]: