

Answers to assignment 6 and 7

Assignment 6

Analysis Problems

For these problems, exact answers vary depending on implementation and what is actually counted.

Array: 4 2 8 3 1

Selection sort:

Comparisons: $4 + 3 + 2 + 1 = 10$

Swaps: $4(N - 1)$ --- may be less depending on implementation

Insertion sort

Comparisons: $1 + 1 + 3 + 4 = 9$

Swaps: $1 + 0 + 2 + 4 = 7$

Array: 1 3 2 4 5

Selection sort:

Comparisons: $4 + 3 + 2 + 1 = 10$

Swaps: $4(N - 1)$ --- may be less depending on implementation

Insertion sort

Comparisons: $1 + 2 + 1 + 1 = 5$

Swaps: $0 + 1 + 0 + 0 = 1$

Note that the insertion sort has fewer operations for the nearly sorted array.

Assignment 7

Problems

1. For each of the following sequences of float values, indicate whether they form an array-based min-heap, max-heap or no heap:
 - a) 4.3 2.1 2.3 1.1 1.9 2.2 --- MAX-HEAP
 - b) 2.3 5.3 4.1 4.7 4.9 7.8 --- NO HEAP

c) 5.7 4.3 4.7 3.0 3.3 4.5 4.6 1.1 2.0 3.1 --- MAX-HEAP

d) 2.3 4.1 5.3 6.7 6.0 5.9 5.4 7.3 7.0 8.3 7.4 6.1 --- MIN-HEAP

e) 7.4 5.3 5.5 4.3 5.1 5.4 --- MAX-HEAP

2. For each of the following sequences of float values, produce the min-heap array that would result if they were inserted in the order that they appear below:

a) 5.4 4.3 6.7 1.2 4.5 7.2 1.1

After inserting in a min-heap → 1.1 4.3 1.2 5.4 4.5 7.2 6.7

b) 3.4 4.3 1.8 1.3 1.9 1.1

After insert in a min-heap → 1.1 1.8 1.3 4.3 1.9 3.4

3. Consider the follow array-based min-heap: 1.3 2.7 2.1 3.2 3.1 6.8 6.7 4.8 5.9 8.3. Produce the min-heap array that would result by removing the minimal element from the heap.

→ 2.1 2.7 6.7 3.2 3.1 6.8 8.3 4.8 5.9

4. Write a class called HeapCheck that has a method that takes an array of float values and indicates whether the array represents a min-heap, a max-heap or no heap. If all values are the same, your code may simply indicate that it's a min-heap. Your main method should demonstrate that your code works correctly.

Perhaps the simplest approach is to define two separate methods called isMinHeap and isMaxHeap. Skipping the root node, both can run through the array checking if the parent of the node is appropriately less/greater than the current node.