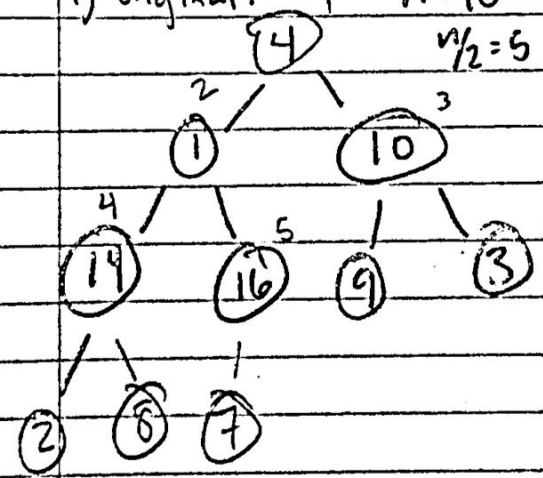
Recitation 3

1) a) Some of the characteristics of a heap data structure is that the either max or min of a group of numbers is at the top of a binary tree. This binary tree is a heap data structure when (in case of max) each child node coming off of the parent is less than or equal to the parent. The opposite being true with a min heap.

Heap Sort running time is $\theta(n \log n)$ because we are extracting each element ($\theta(n)$) and after each extraction we are using the heapify algorithm ($\theta(\log n)$) to re-organize the heap into a proper heap without the previous element. This makes it so we are doing the $\theta(\log n)$ algorithm n times, so $\theta(n \log n)$.

b) Heapify(A, 2)

1) original: n = 10, n/2 = 5

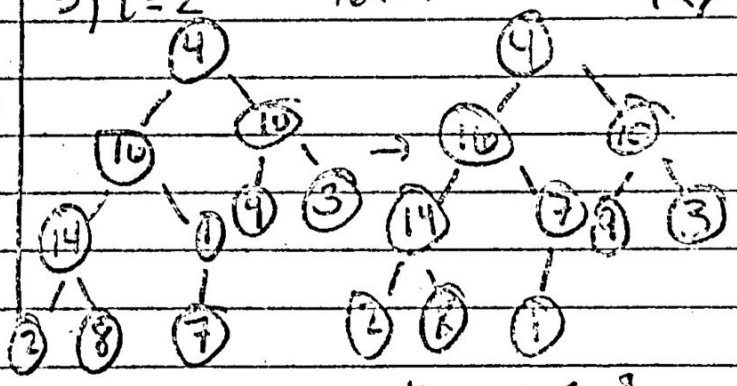


2) i = 5 no change

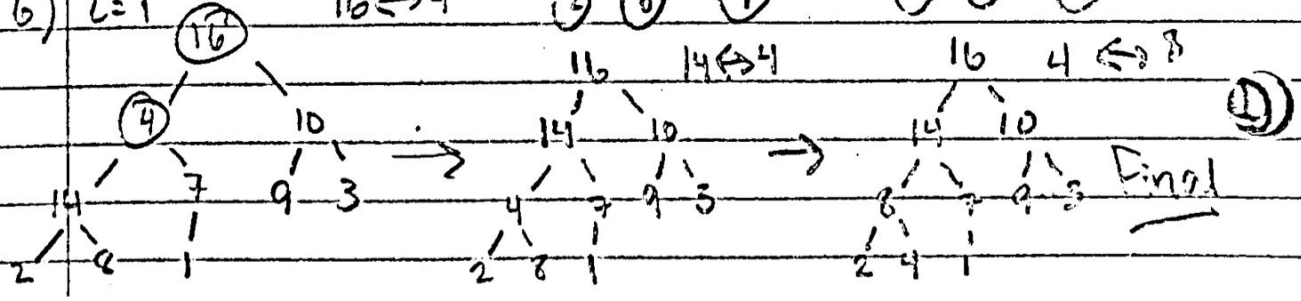
3) i = 4 no change

4) i = 3 no change

5) i = 2 16 ↔ 1 7 ↔ 1



6) i = 1 16 ↔ 4 14 ↔ 4 16 4 ↔ 8



Final

1) c) start: A: {16, 14, 10, 8, 7, 9, 3, 2, 4, 1}

1: A: {1, 14, 10, 8, 7, 9, 3, 2, 4, 16}

heapify:

A: {14, 1, 10, 8, 7, 9, 3, 2, 4, 16}

{14, 8, 10, 1, 7, 9, 3, 2, 4, 16}

{14, 8, 10, 4, 7, 9, 3, 2, 1, 16}

2: A: {1, 8, 10, 4, 7, 9, 3, 2, 14, 16}

heapify:

A: {10, 8, 1, 4, 7, 9, 3, 2, 14, 16}

{10, 8, 9, 4, 7, 1, 3, 2, 14, 16}

Done with 2 iterations.

2) See C++ program.