Ethan Gutknecht
5/6/2021

1. Heap sort works by taking a max heap and sorting it using a heap's properties. The first step of that is taking the max element, index 1, or the root of the tree and switch it with the end of the array. After that, we heapify the array giving the properties of that heap back to the array. We continue to do this process until the ending pointer reaches the front of the array or there is one element in the unsorted part of the array. A demonstration of this is below.

	Array	**Step**
1	{12, 6, 10, 5, 1, 9,}	Heapify
2	{9, 6, 10, 5, 1, 12}	Switch front and last
3	{10, 6, 9, 5, 1, 12}	Heapify
4	{1, 6, 9, 5, 10, 12}	Switch front and last
5	{9, 6, 1, 5, 10, 12}	Heapify
6	{5, 6, 1, 9, 10, 12}	Switch front and last
7	{6, 5, 1, 9, 10, 12}	Heapify
8	{1, 5, 6, 9, 10, 12}	Sorted

	Array	**Step**
1	{1, 14, 7, 8, 3}	Heapify
2	{3, 8, 7, 1, 14}	Switch front and last
3	{8, 3, 7, 1, 14}	Heapify
4	{1, 3, 7, 8, 14}	Switch front and last
5	{7, 3, 1, 8, 14}	Heapify
6	{1, 3, 7, 8, 14}	Switch front and last
7	{1, 3, 7, 8, 14}	Sorted

2. I would say no. This would technically count as sorting data and the lowest cost to sort data is nlog(n).