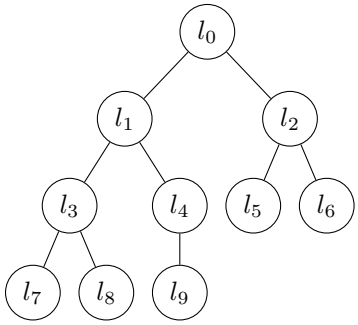
Sorting

$$l = (l_0, l_1, l_2, \ldots, l_{n-1})$$
$$l_0 \leq l_1 \leq \cdots \leq l_{n-1}$$

On a tree



Heaps

Heap: complete binary tree.

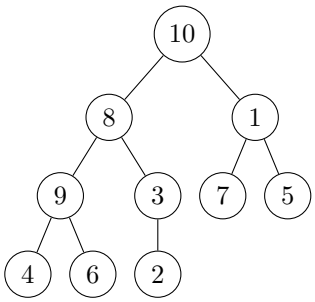
Max heap: each parent bigger than children.

Min heap: each parent smaller than children.

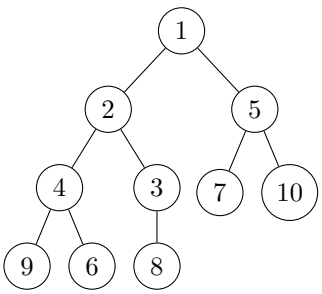
Check for min/max heap in $(n-1)$ comparisons.

To max/min heap

1. Start with last node, moving backwards.
2. Compare node to children, swap if needed.
3. Swap parent down tree until we have a heap.



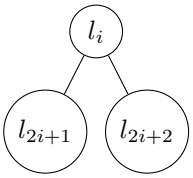
Last five nodes have no children. Sixth-last has one child, is bigger so swap. Now have a heap from sixth-last. Same for seventh-last: swap 9 for 4. Third node is a heap. Second node swaps 2 for 8, and filters down swapping 3 for 8. Finally, the root is swapped with 1 and then 5. This gives:



Heap Sort

1. Convert complete binary tree to heap.
2. Remove root, insert into new list.
3. Remove last element, place at root.
4. Repeat until empty.

As an array



(	l_0,	l_1,	l_2,	l_3,	l_4,	l_5,	l_6,	l_7,	l_8,	l_9)
(	10,	8,	1,	9,	3,	7,	5,	4,	6,	2)
(	10,	8,	1,	9,	2,	7,	5,	4,	6,	3)
(	10,	8,	1,	4,	2,	7,	5,	9,	6,	3)
(	10,	2,	1,	4,	8,	7,	5,	9,	6,	3)
(	10,	2,	1,	4,	3,	7,	5,	9,	6,	8)
(	1,	2,	10,	4,	3,	7,	5,	9,	6,	8)
(	1,	2,	5,	4,	3,	7,	10,	9,	6,	8)