HEAPS and HEAP-SORT

specifically binary MAX-heaps
Rules:
- binary
- max
- complete
Rules:

- **binary**: internal nodes have 1 or 2 children
- **max**
- **complete**
Rules:
- binary: internal nodes have 1 or 2 children
- max: parent $\geq$ child
- complete
Rules:

- **binary**: internal nodes have 1 or 2 children
- **max**: parent $\geq$ child
- **complete**: all levels filled (lowest can be partial, left to right)
Rules:

- *binary*: internal nodes have 1 or 2 children
- *max*: parent ⩾ child
- *complete*: all levels filled (lowest can be partial, left to right)

Some applications don't need this but we will enforce it
[Notice every subtree is also a heap]

Rules:

- **binary**: internal nodes have 1 or 2 children
- **max**: parent ≥ child
- **complete**: all levels filled (lowest can be partial, left to right)

Some applications don't need this but we will enforce it
How can we identify the indices of the children of a given node?
How can we identify the indices of the children of a given node?

\[
\text{left-child}(i) = 2i \\
\text{right-child}(i) = 2i + 1
\]
How can we identify the indices of the children of a given node?

left-child(i) = 2i

right-child(i) = 2i + 1

parent(i) = ?
How can we identify the indices of the children of a given node?

left-child(i) = 2i
right-child(i) = 2i + 1
parent(i) = \( \lfloor i/2 \rfloor \)
How can we identify the indices of the children of a given node?

Use array to store heap (avoid wasting space with pointers)

\[
\begin{align*}
\text{left-child}(i) &= 2i \\
\text{right-child}(i) &= 2i + 1 \\
\text{parent}(i) &= \left\lfloor \frac{i}{2} \right\rfloor
\end{align*}
\]
How does this relate to sorting?
How does this relate to sorting?

Largest element is on top.
How does this relate to sorting?

Largest element is on top.

2nd largest is in level 2.
How does this relate to sorting?

Largest element is on top.
2nd largest is in level 2.
3rd largest is \( 5 \) in level 2.
How does this relate to sorting?

Largest element is on top.
2nd largest is in level 2.
3rd largest is
  b in level 2
  or
  4 in level 3
  & child of 2nd

getting messy
Heaps are not "sorted"

How does this relate to sorting?

Largest element is on top.
2nd largest is in level 2.
3rd largest is
- in level 2
- or
- in level 3
& child of 2nd

Getting messy
How to sort data in a heap
How to sort data in a heap

extract max

16

(copy to output array)
How to sort data in a heap

Update max: larger of 2 children
How to sort data in a heap
How to sort data in a heap

Update max recursively

14
    /   \
   8    10
    /   /  \
   2   7   9
    / \
   4   3

16
How to sort data in a heap
How to sort data in a heap
How to sort data in a heap
How to sort data in a heap

If we don't care about keeping the heap complete
How to sort data in a heap

- if we don't care about
  - keeping the heap complete
  - using extra space

(output array)
How to sort data in a complete heap ... using extra space
How to sort data in a complete heap ... using extra space

- Extract max

Diagram:
- 16
- 14
- 10
- 8
- 7
- 9
- 3
- 2
- 4
- 1
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored
How to sort data in a **complete** heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored

```
  14
   |  
  8   10
 |    |  
 7   9  3
  |  
 2   4
```

"heapify"
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored

time?
How to sort data in a complete heap ... using extra space

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored

time?

$O(\log n)$ per extraction
How to sort data in a complete heap ... using extra space

16

- extract max
- replace root with rightmost leaf from lowest level
- recursively swap with largest child while heap not restored

time = O(nlogn)
0(logn) per extraction
How to sort data in a complete heap in place (without an output array)
How to sort data in a complete heap in place (without an output array)
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before
but we swap
max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

start heapifying

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

Make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive as though extracted

valid heap
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

1 2 3 4 5 6 7 8 9
14 8 10 4 7 9 3 2 1 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

1 2 3 4 5 6 7 8 9
1 8 10 4 7 9 3 2 14 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive

1 2 3 4 5 6 7 8

1 8 10 4 7 9 3 2 14 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive

1 2 3 4 5 6 7 8

1 8 10 4 7 9 3 2 14 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive

1 2 3 4 5 6 7 8 10 8 1 4 7 9 3 2 14 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive

10 8 9 4 7 1 3 2 14 16
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

make replacement position inactive

valid heap
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement.
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

```
1 2 3 4 5 6 X X X X
8 7 3 4 2 1 9 10 14 16
```
How to sort data in a complete heap in place (without an output array)

Same as before but we swap max with replacement

e tc
Summary

Given a heap we can extract max and heapify in $O(\log n)$ time.

$n$ rounds: $O(n\log n)$ to sort a heap
Summary

Given a heap we can extract max and heapify in $O(\log n)$ time.

$n$ rounds: $O(n \log n)$ to sort a heap

How do we construct a heap in the first place?