File Edit View Help

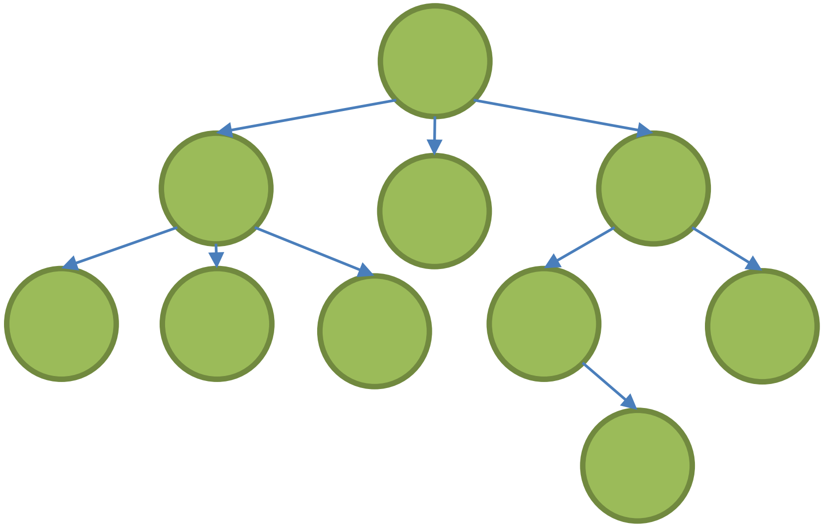
File Edit View Help

Heaps

Trees

- Definition: A data structure that can be defined recursively as a collection of nodes, where each node is a data structure consisting of a value, together with a list of references (edges) to nodes, with the constraints that no reference is duplicated, and none points to the root.

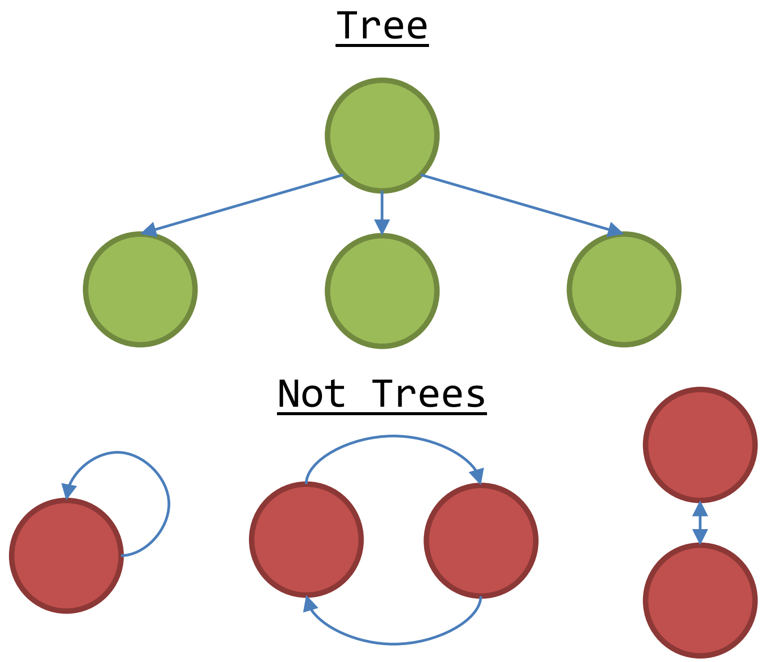
Trees



File Edit View Help

Trees

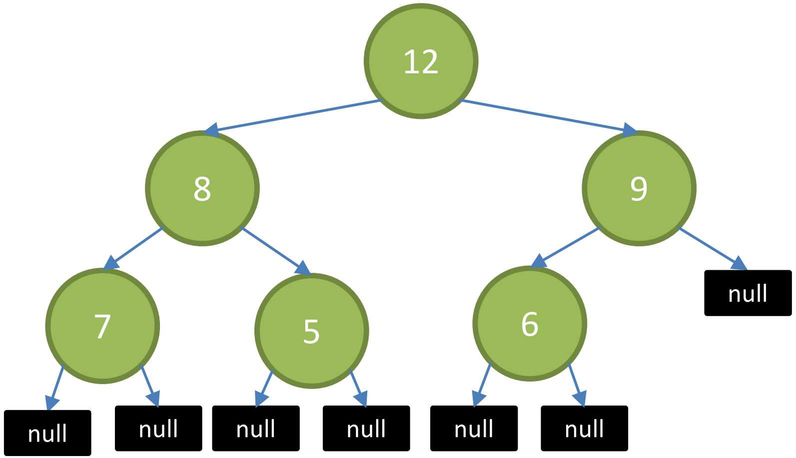
- Trees Have
  - Nodes
  - Edges
- Trees CANNOT
  - Contain Self-Referencing Edges
  - Have Cycles
  - Be Disjointed



Heaps

- Binary Tree Structure
- Node’s data must be comparable
- Node’s have at most two children
  - Left Child
  - Right Child
- Max Heap: Children must be less than or equal to the parent
- Min Heap: Children must be greater than or equal to the parent
- Assume Leaves are NULL references

Max Heap



File Edit View Help

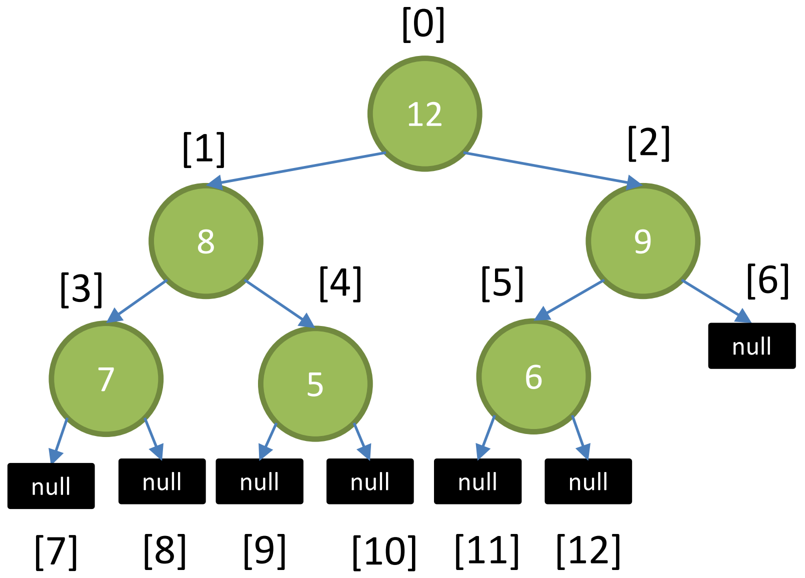
Heaps

- Array Heap
- Assume Root is at Index 0
- Left Child Index = Parent Index * 2 + 1
- Right Child Index = Parent Index * 2 + 2
- Parent Index = (Child Index-1)/2

Array Max Heap

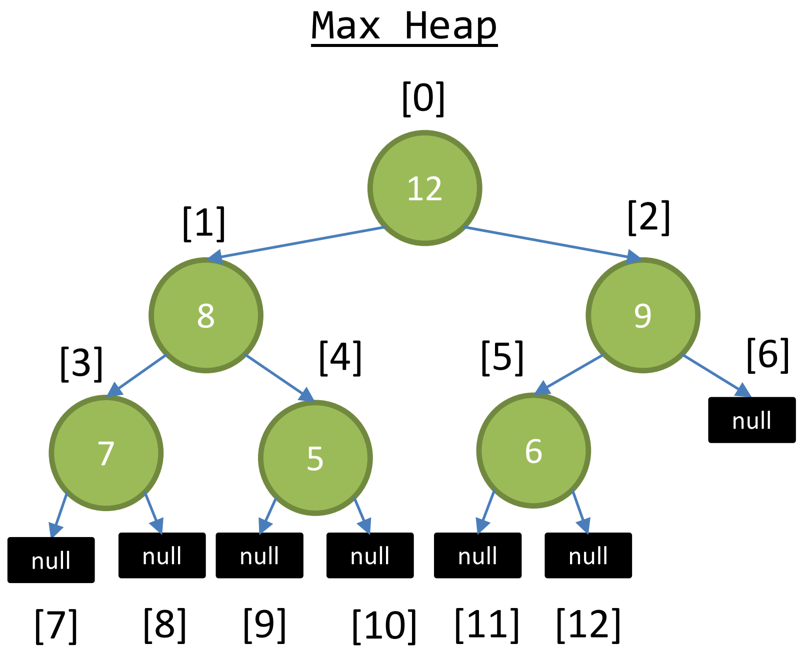
0	1	2	3	4	5	6	7	8	9	10	11	12
12	8	9	7	5	6	-	-	-	-	-	-	-

Max Heap



Heaps

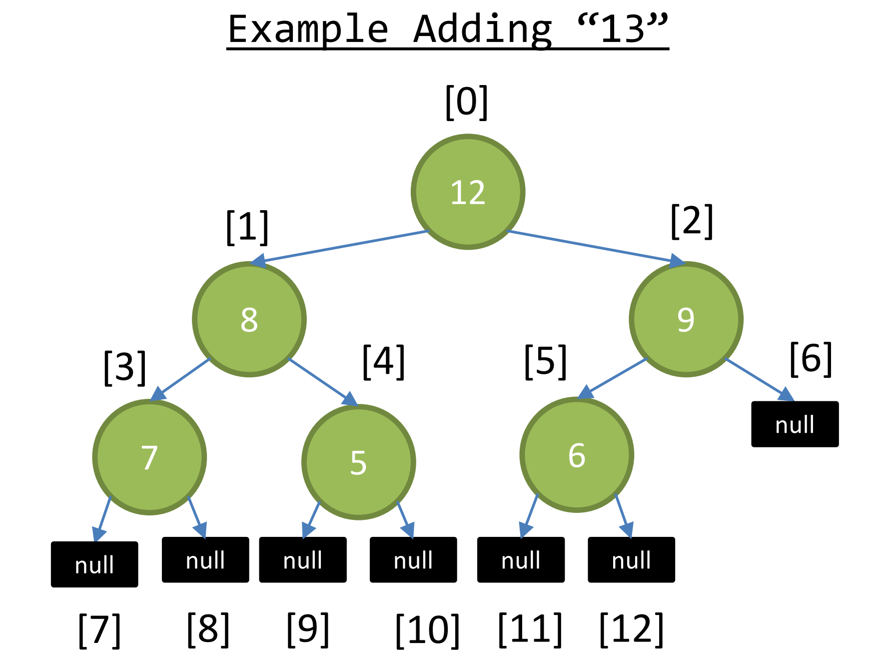
- Add
  - Replace the first leaf in breadth order with the new data
  - From that node "bubble up" the data if necessary
- Bubble Up
  - If the child's data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index



File Edit View Help

Heaps

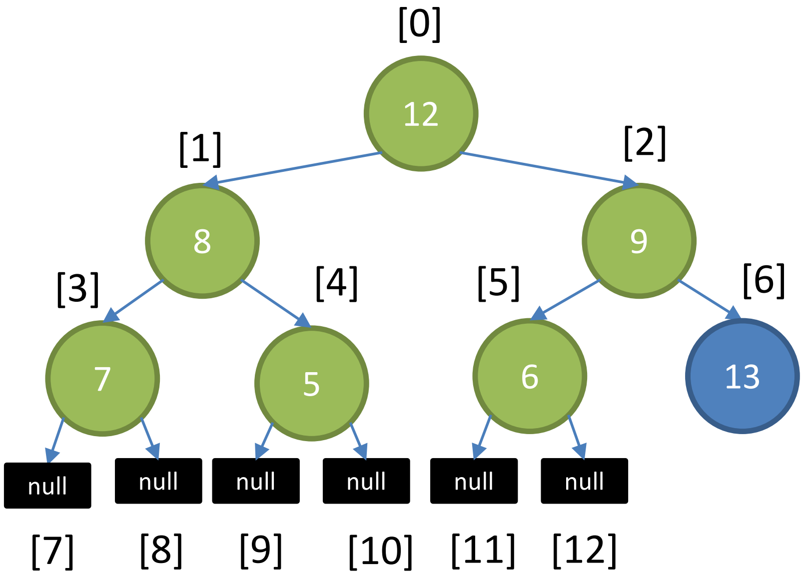
- Add
  - Replace the first leaf in breadth order with the new data
  - From that node "bubble up" the data if necessary
- Bubble Up
  - If the child's data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index



Heaps

- Add
  - Replace the first leaf in breadth order with the new data
  - From that node “bubble up” the data if necessary
- Bubble Up
  - If the child’s data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index

Example Adding “13”

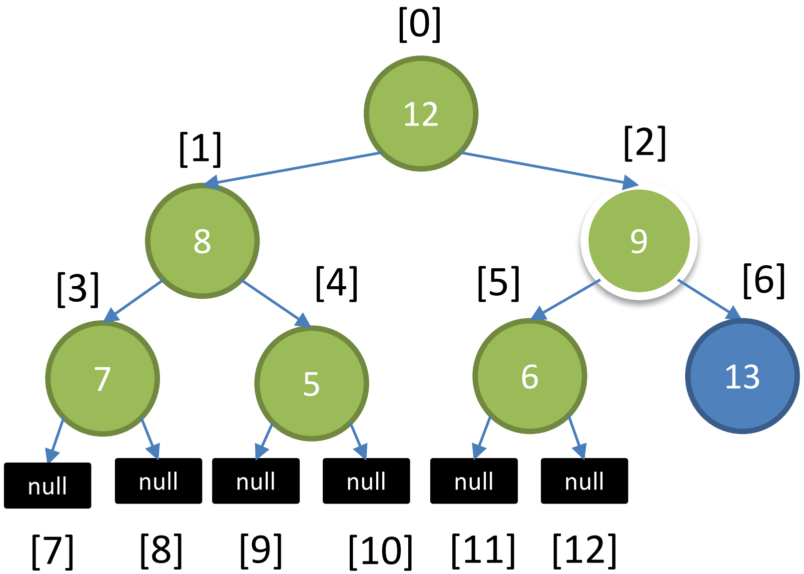


File Edit View Help

Heaps

- Add
  - Replace the first leaf in breadth order with the new data
  - From that node "bubble up" the data if necessary
- Bubble Up
  - If the child's data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index

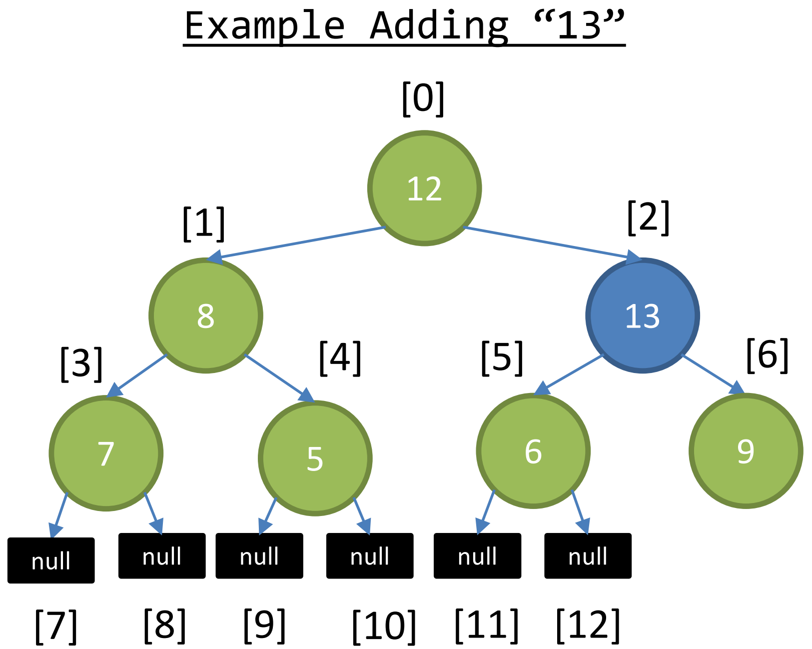
Example Adding "13"



File Edit View Help

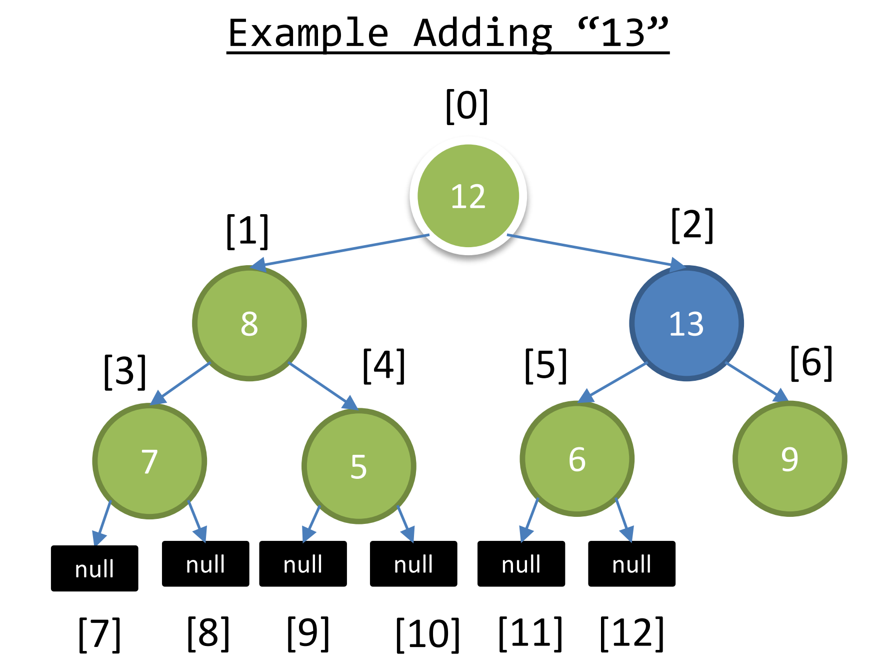
Heaps

- Add
  - Replace the first leaf in breadth order with the new data
  - From that node “bubble up” the data if necessary
- Bubble Up
  - If the child’s data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index



Heaps

- Add
  - Replace the first leaf in breadth order with the new data
  - From that node "bubble up" the data if necessary
- Bubble Up
  - If the child's data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index

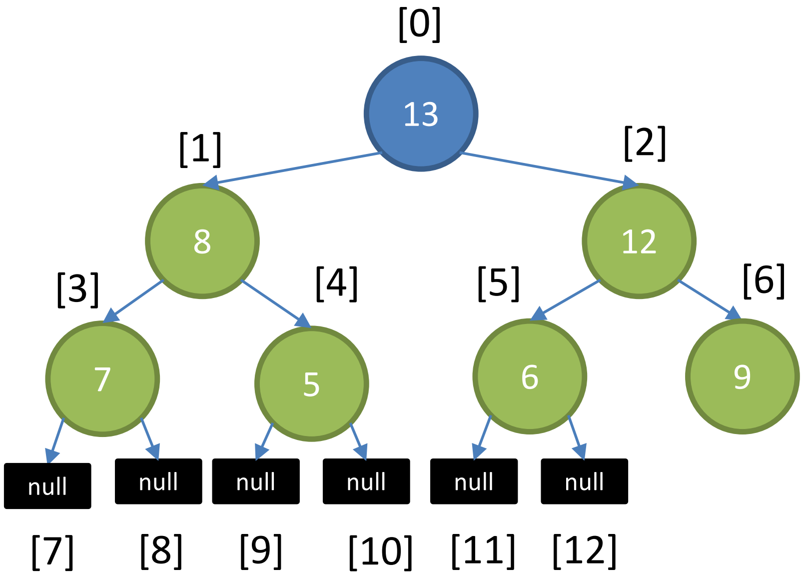


File Edit View Help

Heaps

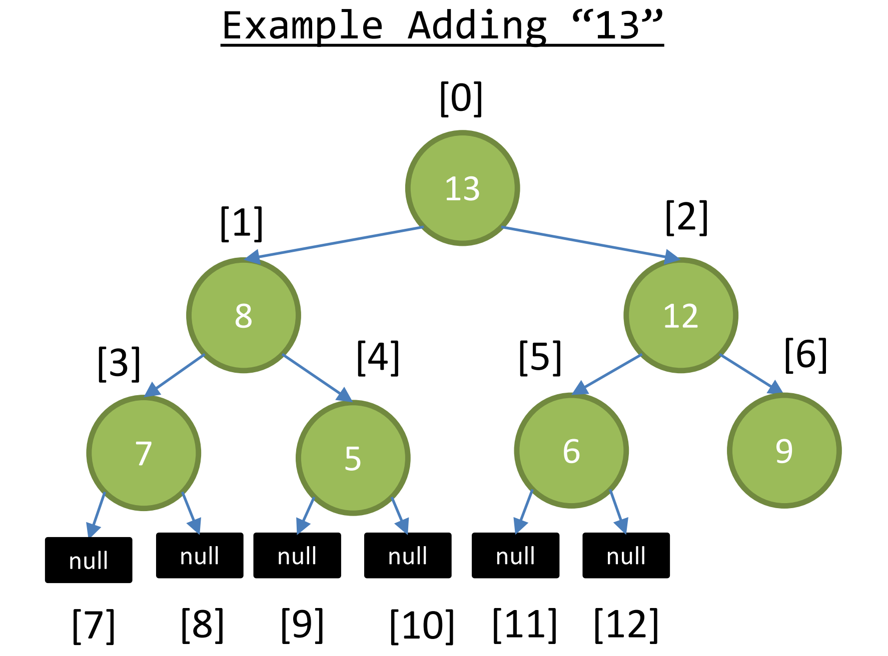
- Add
  - Replace the first leaf in breadth order with the new data
  - From that node “bubble up” the data if necessary
- Bubble Up
  - If the child’s data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index

<u>Example Adding “13”</u>



Heaps

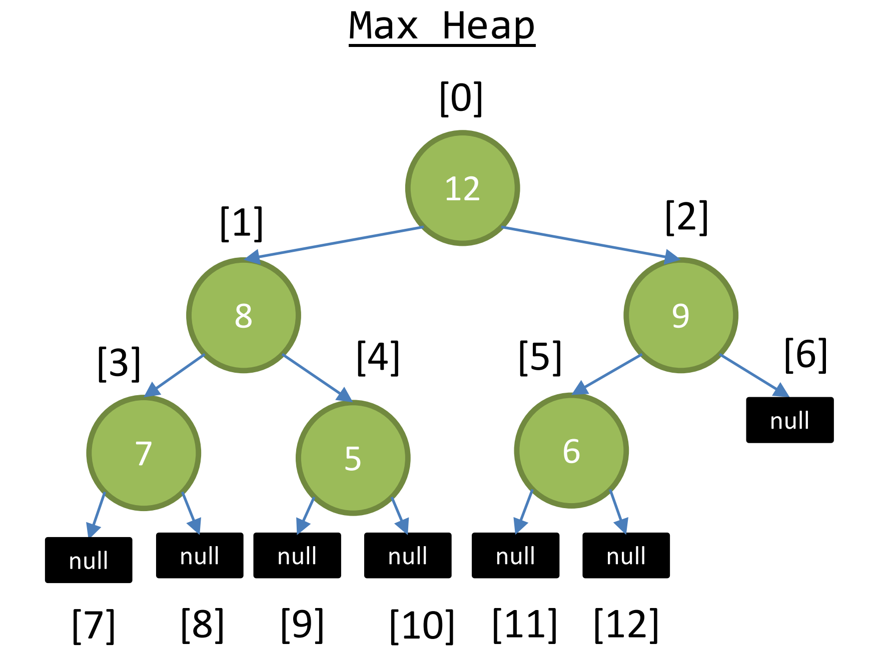
- Add
  - Replace the first leaf in breadth order with the new data
  - From that node "bubble up" the data if necessary
- Bubble Up
  - If the child's data is larger than the parent then swap that information
  - Continue swapping child data with parent data until the parent is larger than the child or we reach the root index



File Edit View Help

Heaps

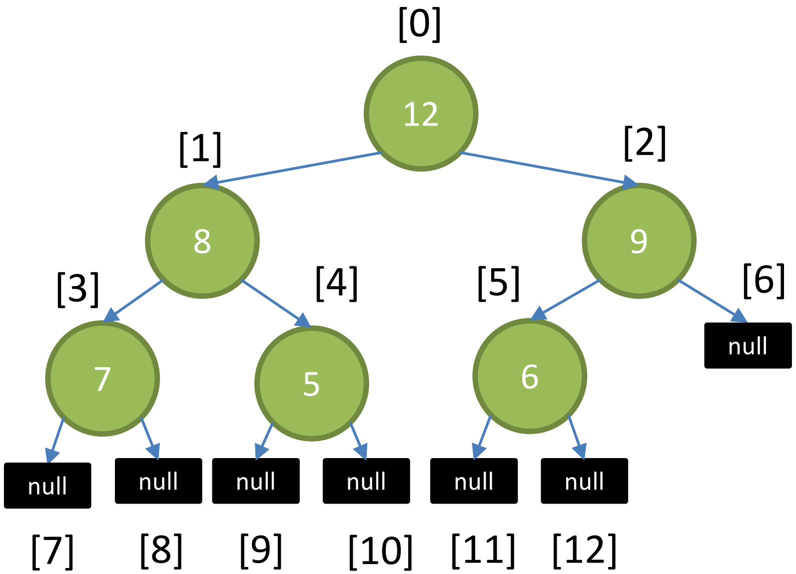
- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds



Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

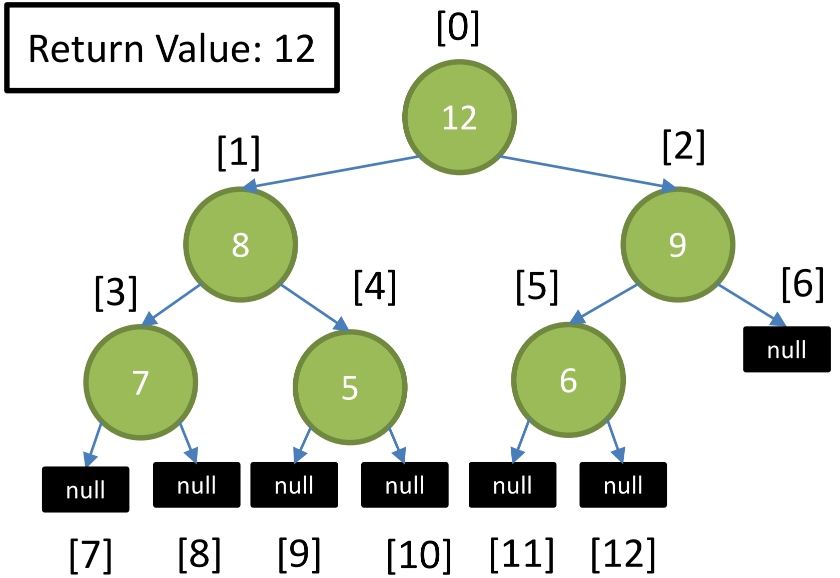
Example Remove



Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

<u>Example Remove</u>



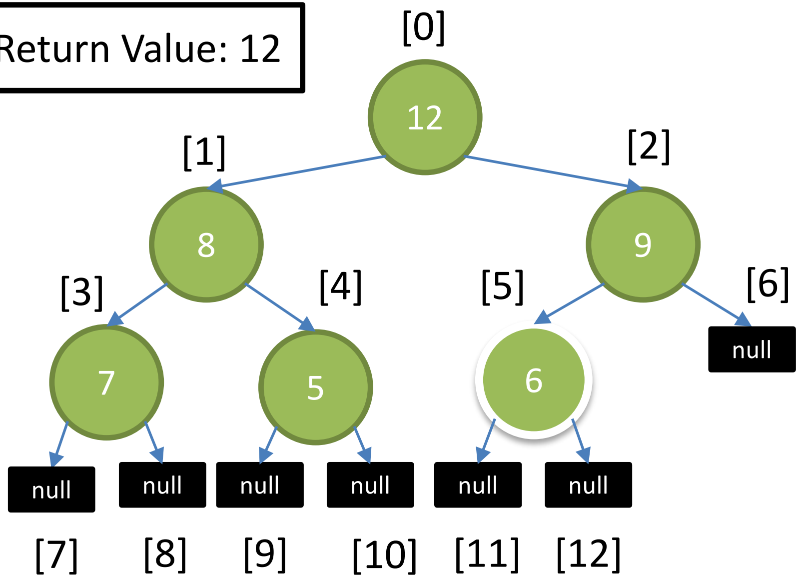
File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

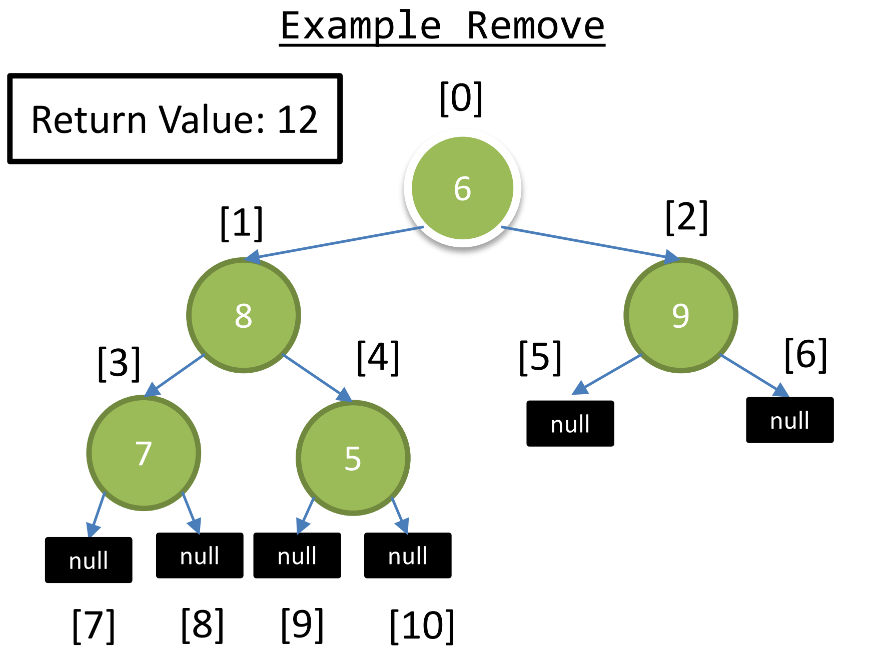
Example Remove

Return Value: 12



Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds



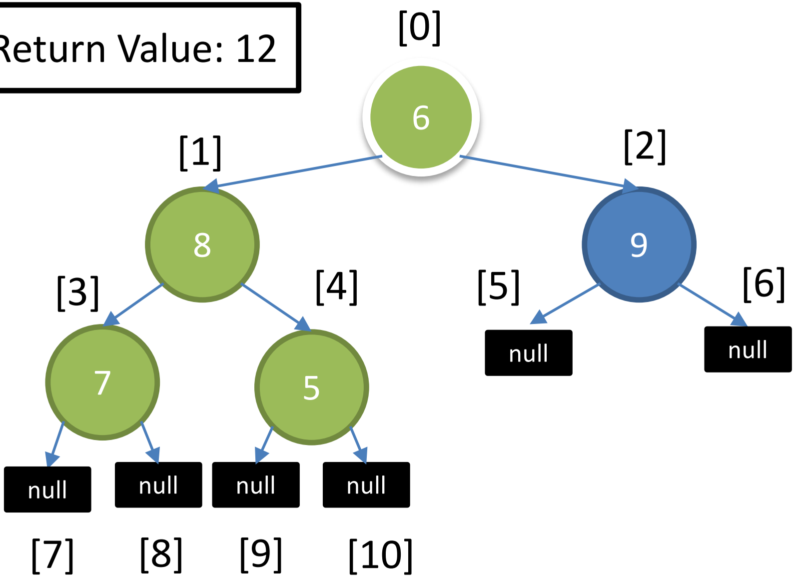
File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

<u>Example Remove</u>

Return Value: 12

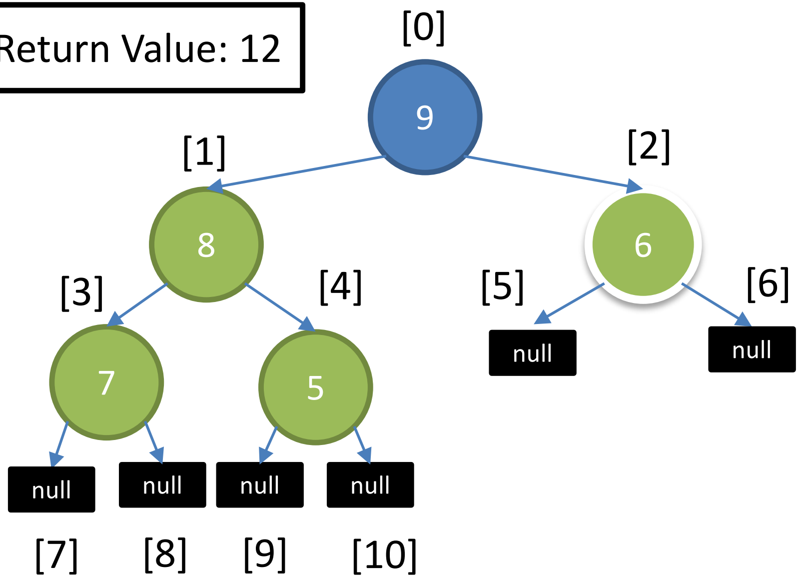


Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

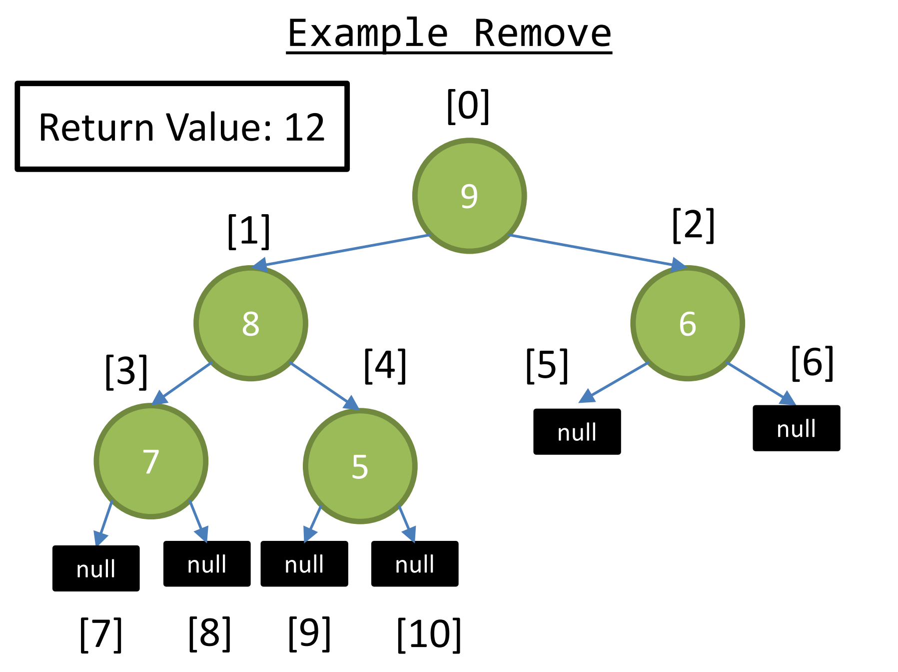
Example Remove

Return Value: 12



Heaps

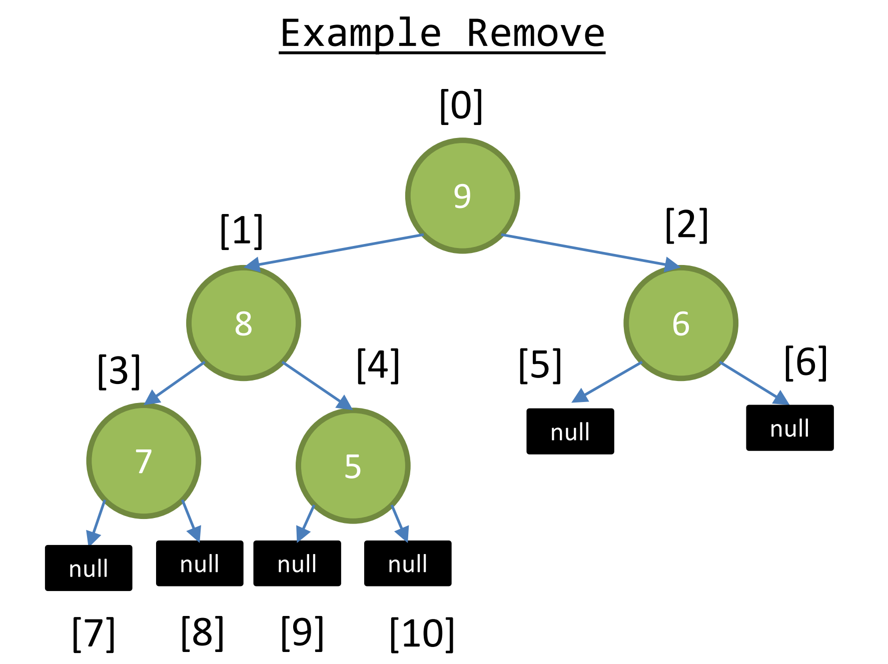
- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds



File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds



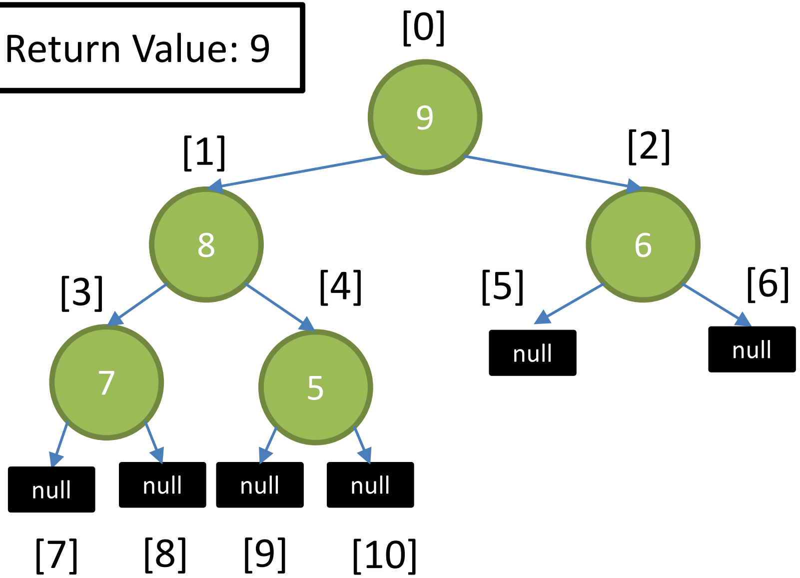
File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again

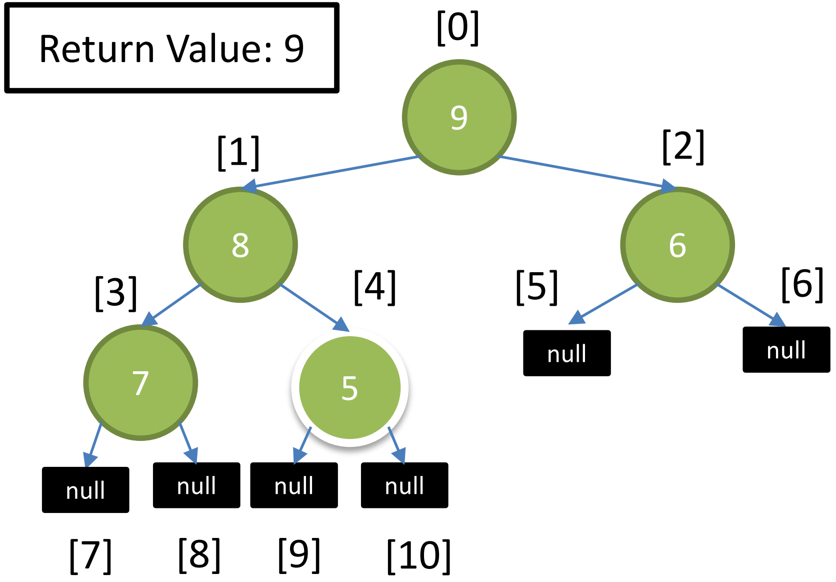
Return Value: 9



Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again

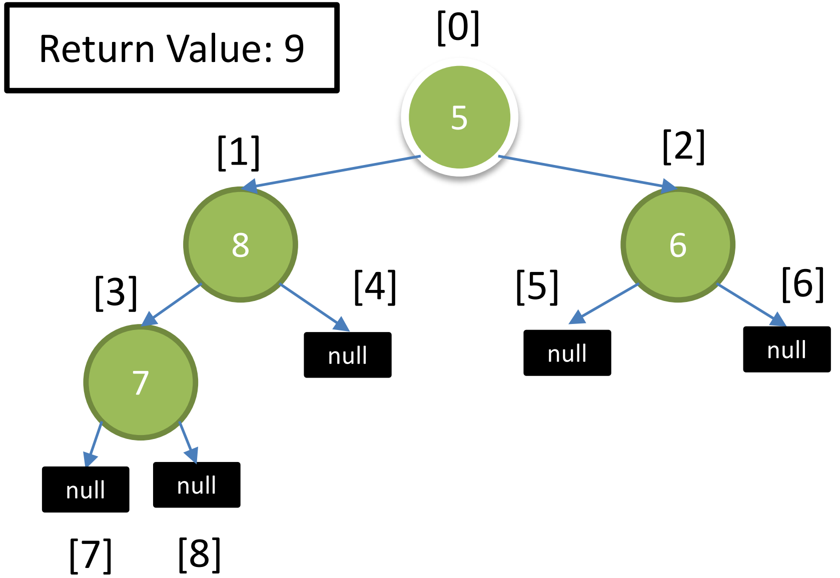


File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again



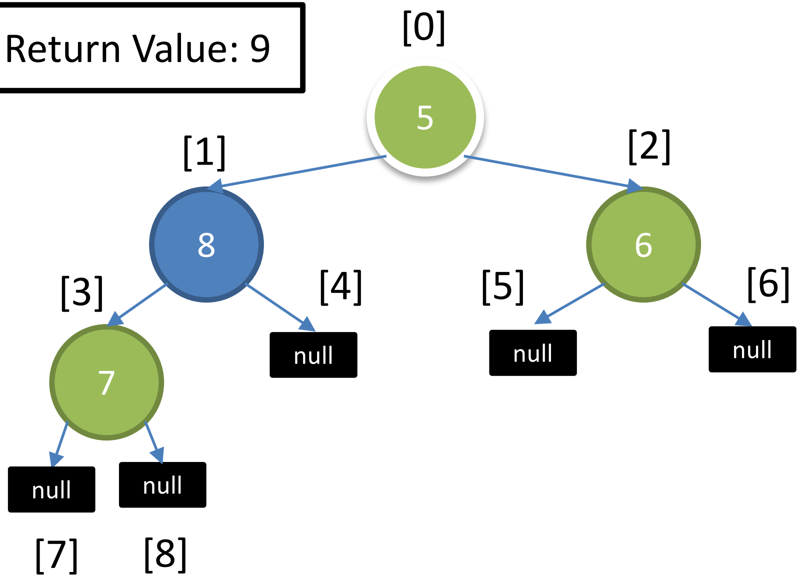
File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again

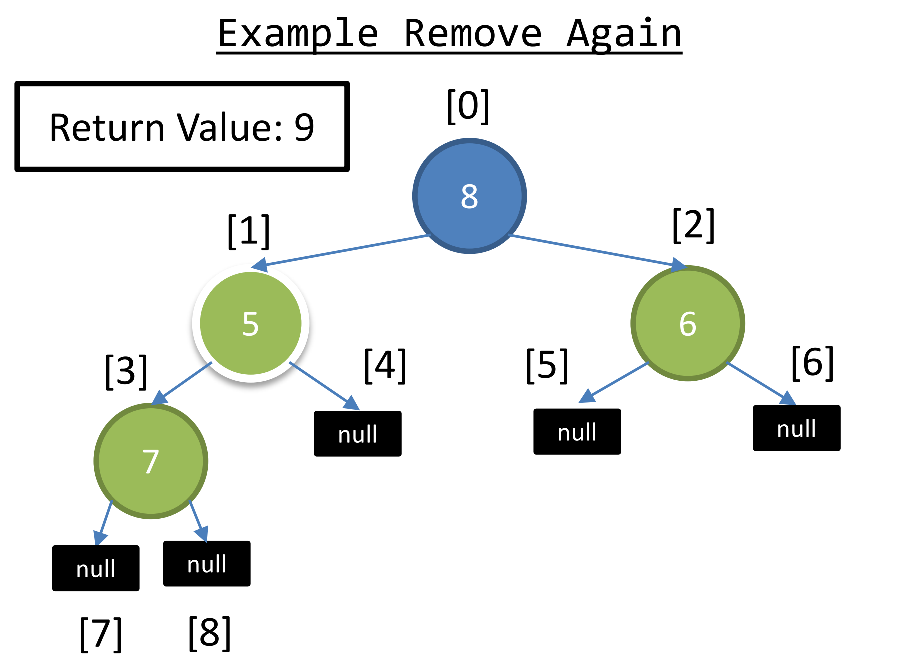
Return Value: 9



File Edit View Help

Heaps

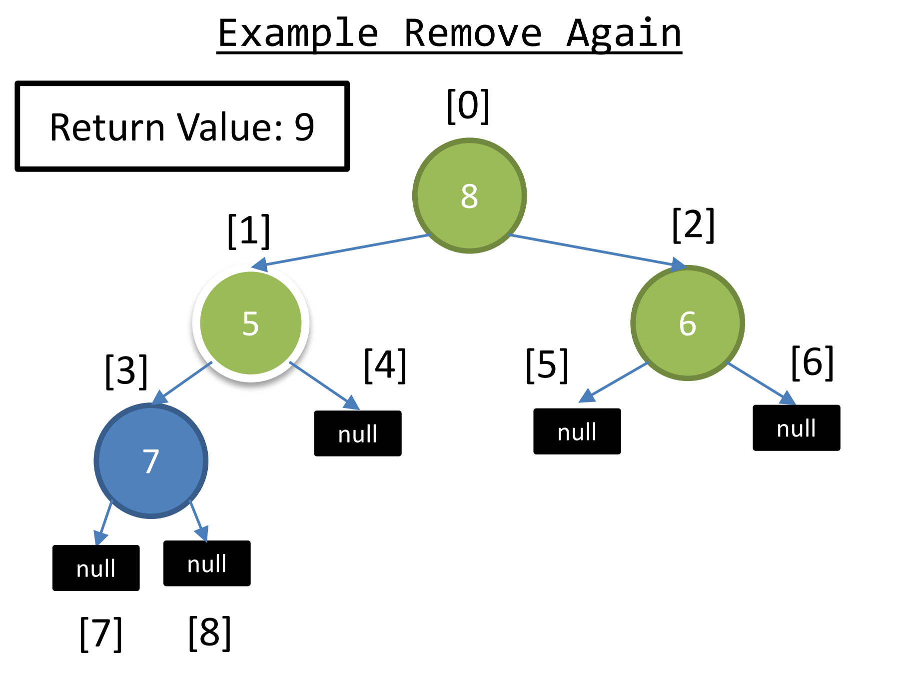
- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds



File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

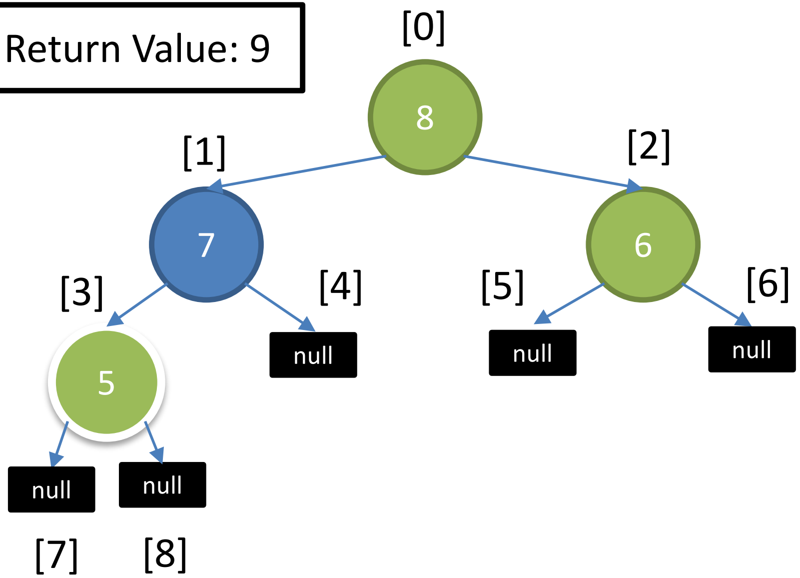


Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again

Return Value: 9



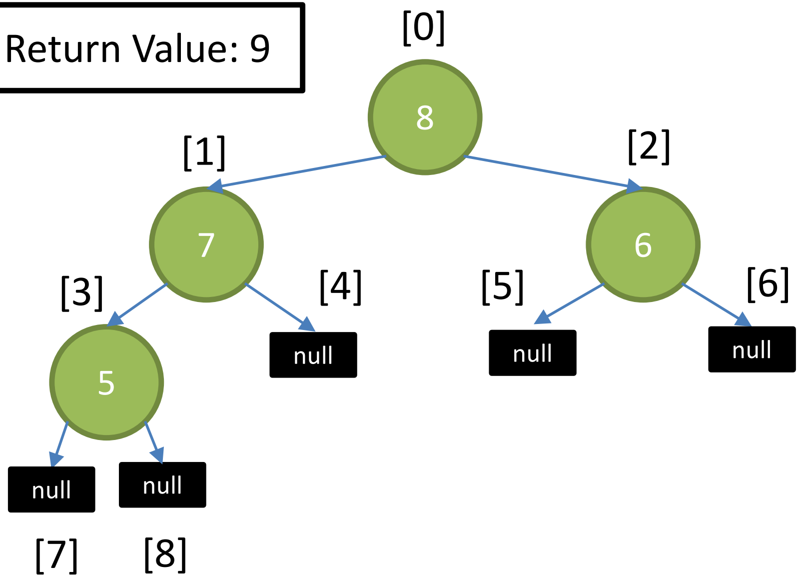
File Edit View Help

Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

<u>Example Remove Again</u>

Return Value: 9



Heaps

- Remove
  - Store the data at the Root
  - Replace the Root data with the Data in the last node in Breadth Order
  - Starting from the root, “Bubble Down” that information
  - Return the stored value, previously at the root
- Bubble Down
  - Pick the larger of the 2 children
  - If its value is larger than the parent, then swap those values
  - Continue this until the parent’s value is larger or we reach the tree’s bounds

Example Remove Again