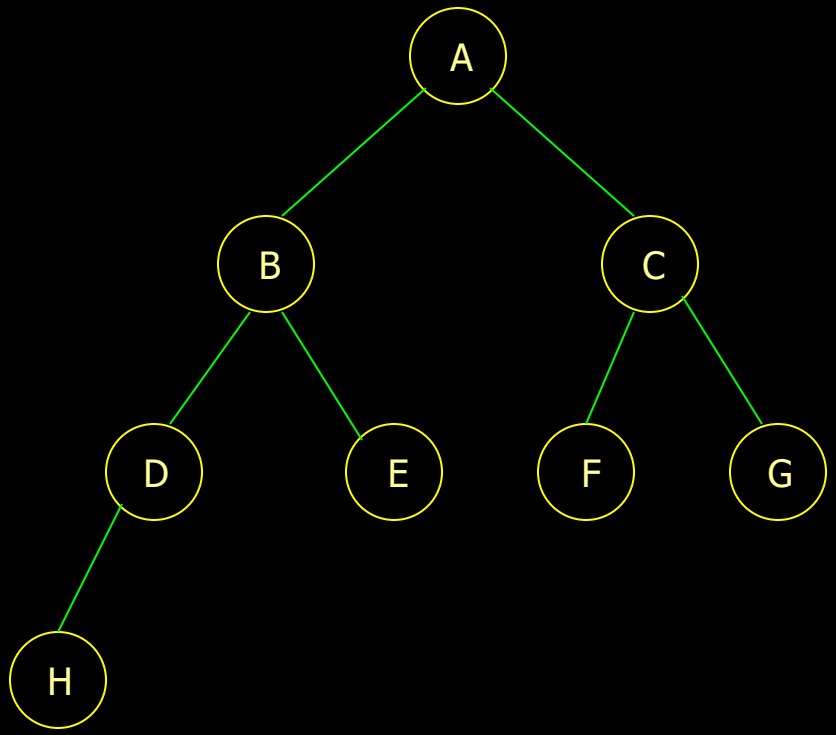
Lecture # 17
Heap

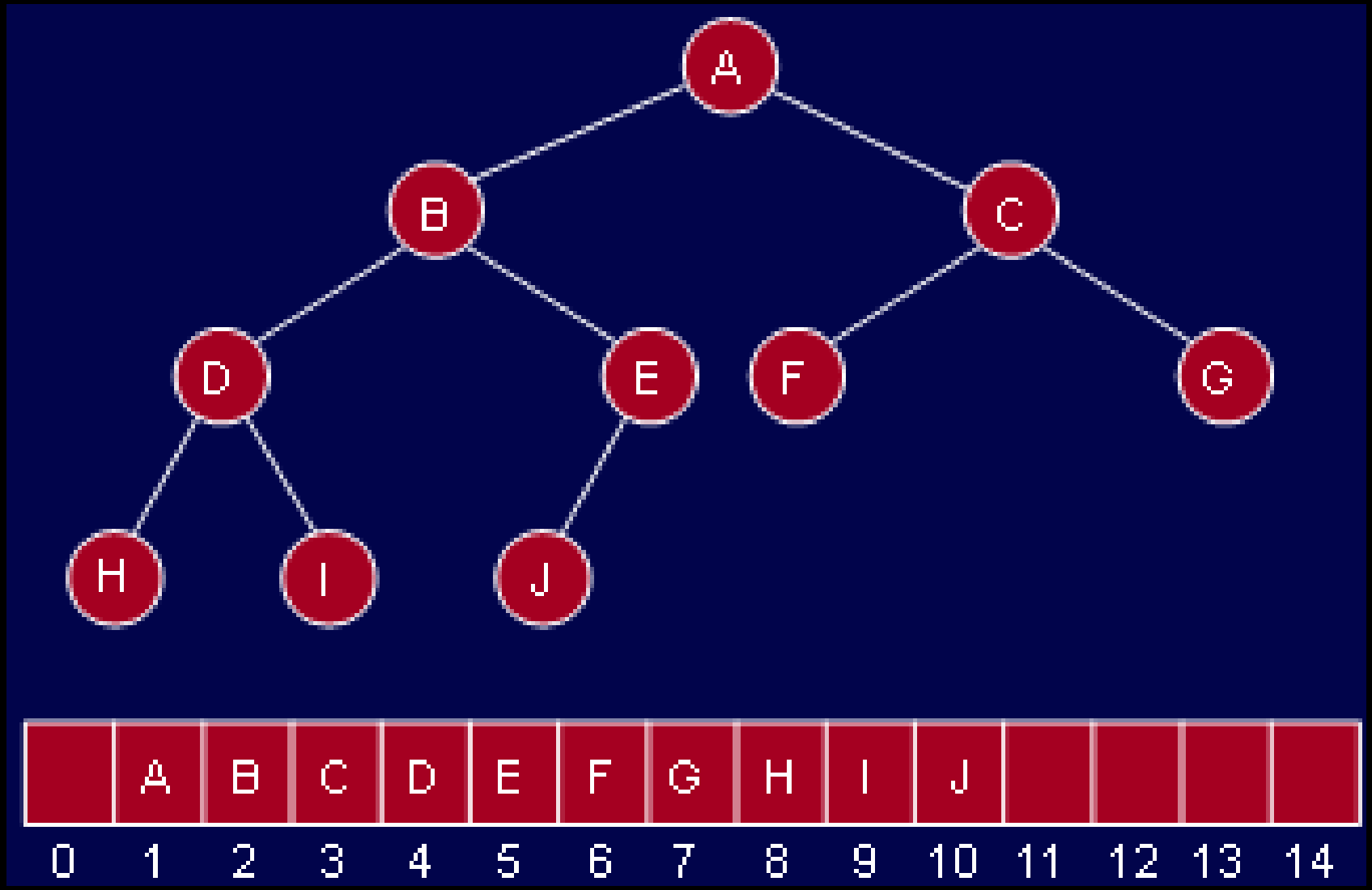
Complete Binary Tree



Complete Binary Tree

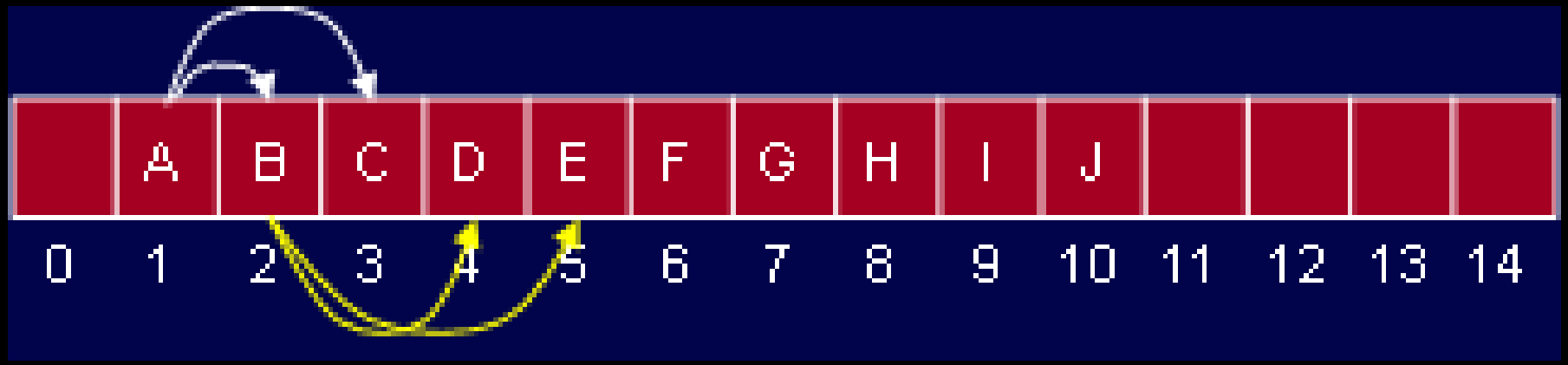
- Recall that such a tree of height h has between 2^h to $2^{h+1} - 1$ nodes.
- Because the tree is so regular, it can be stored in an *array*; no pointers are necessary.

Complete Binary Tree

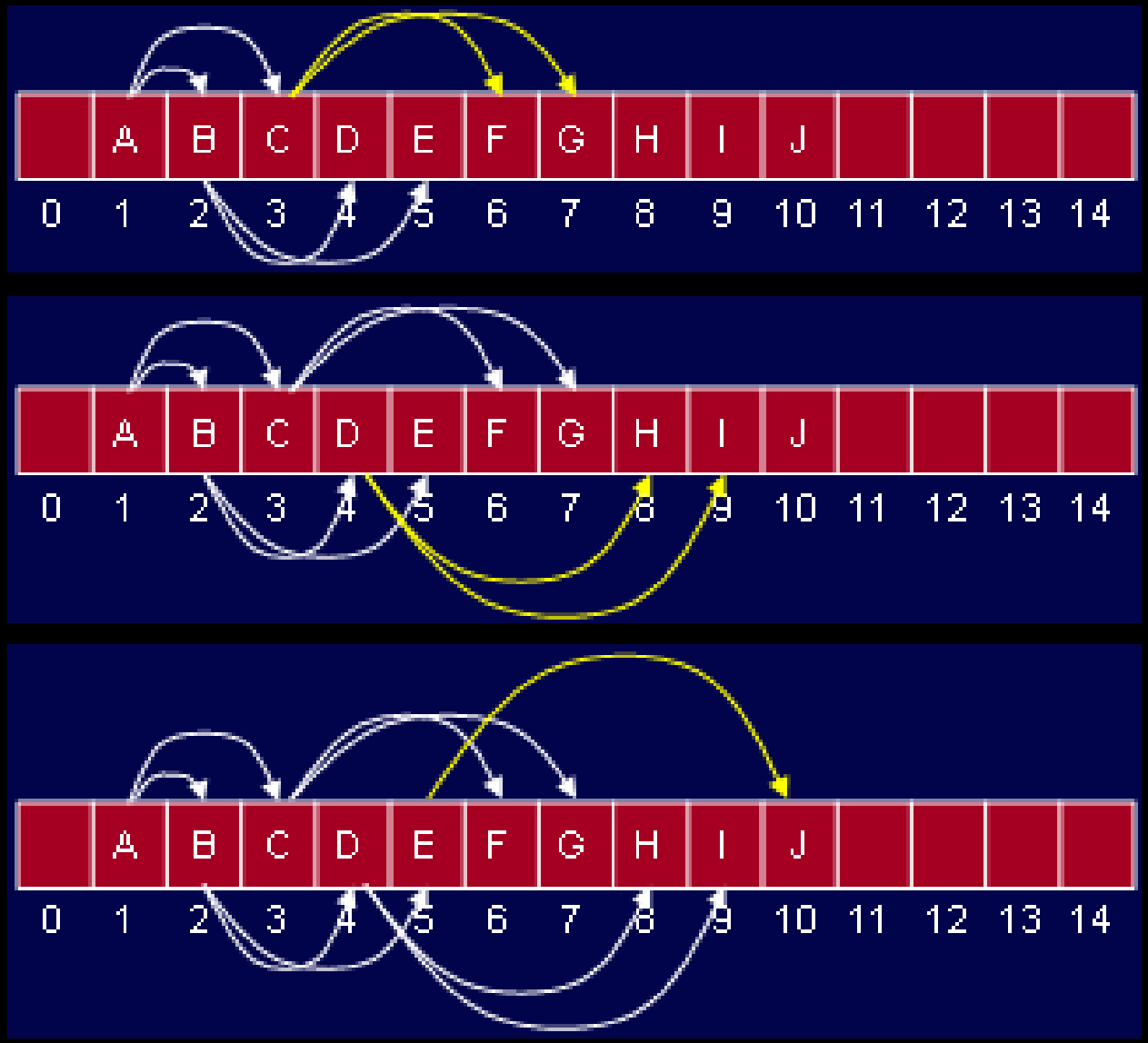


Complete Binary Tree

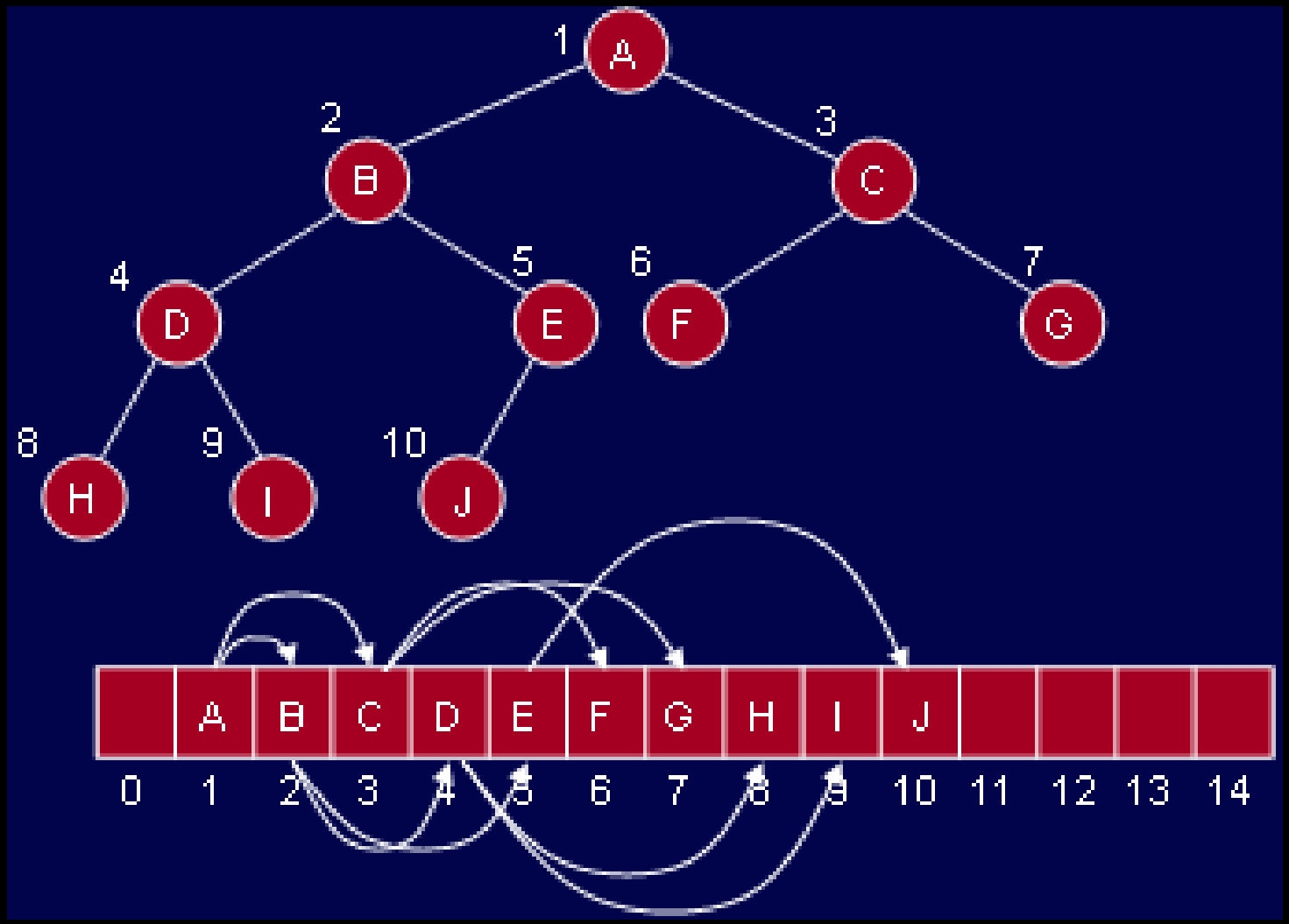
- For any array element at position *i*, the left child is at $2i$, the right child is at $(2i+1)$ and the parent is at $\lfloor i/2 \rfloor$.



Complete Binary Tree



Complete Binary Tree



Complete Binary Tree

- Question:

  why don't we store all binary trees in arrays? Why use pointers?

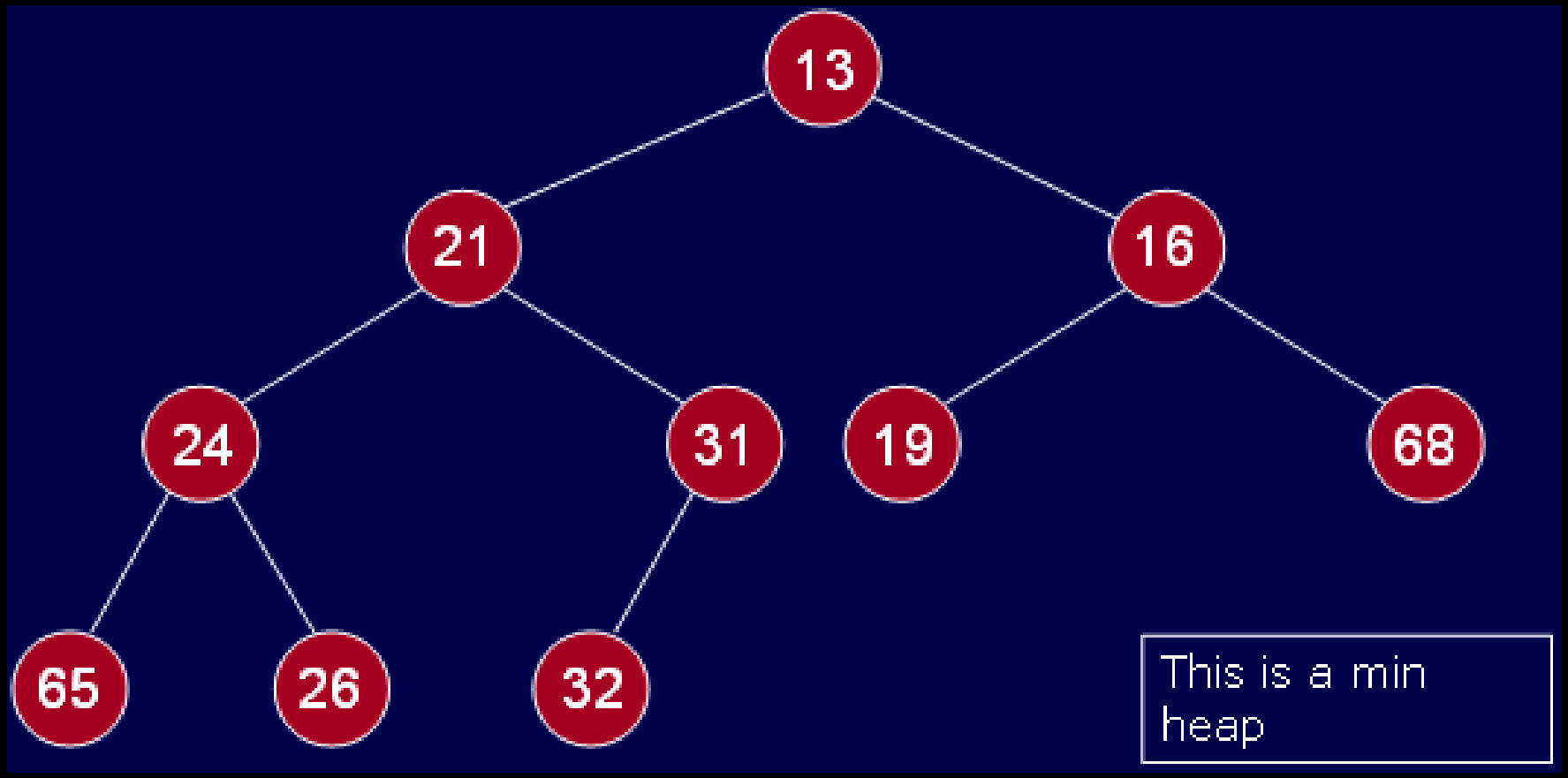
The Heap ADT

- The major usage of heap is in Priority Queues.

Heap

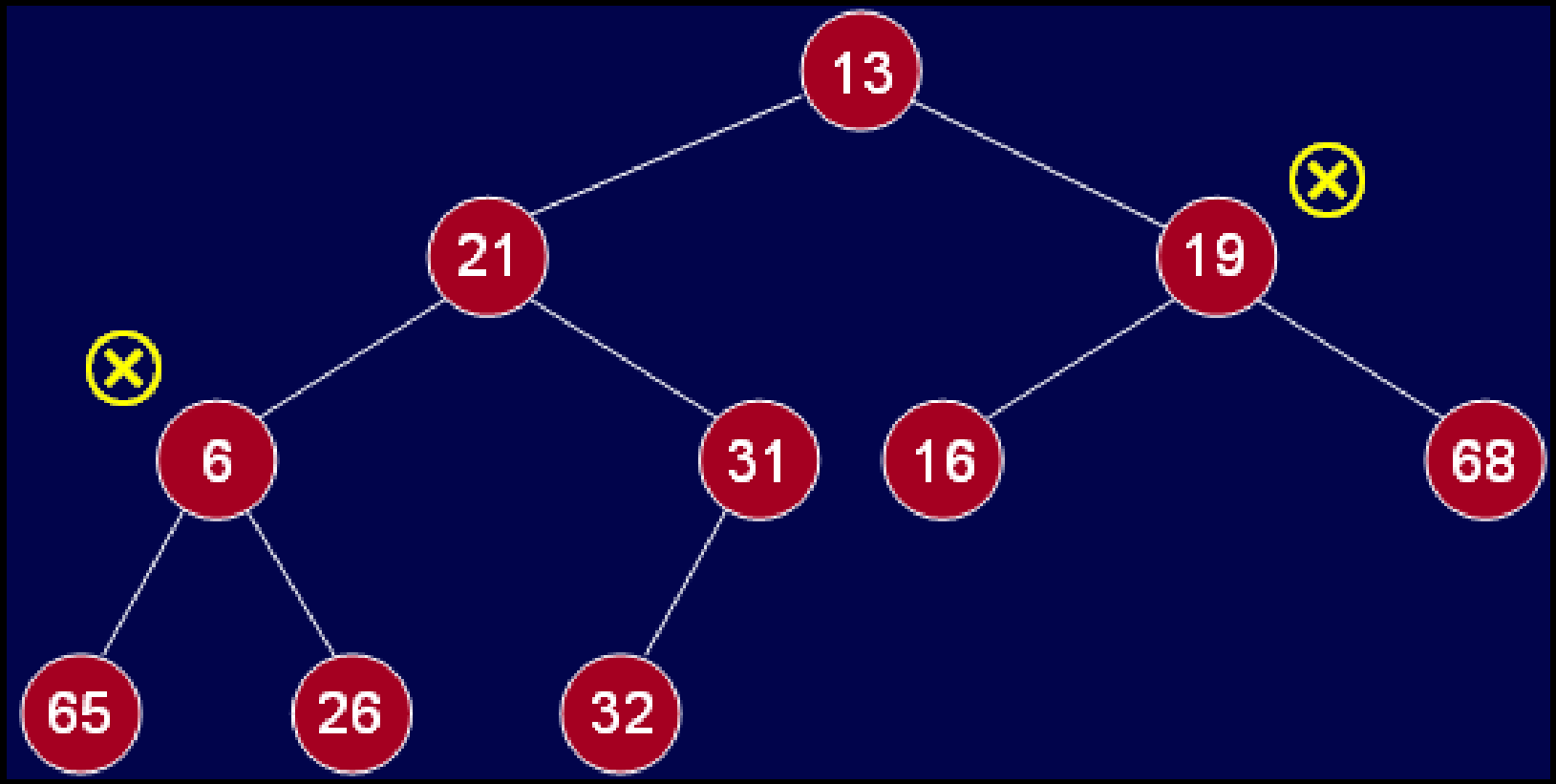
- A heap is a complete binary tree that conforms to the heap order.
- The *heap order* property: in a (min) heap, for every node X, the key in the parent is smaller than (or equal to) the key in X.
- Or, the parent node has key smaller than or equal to both of its children nodes.

Heap



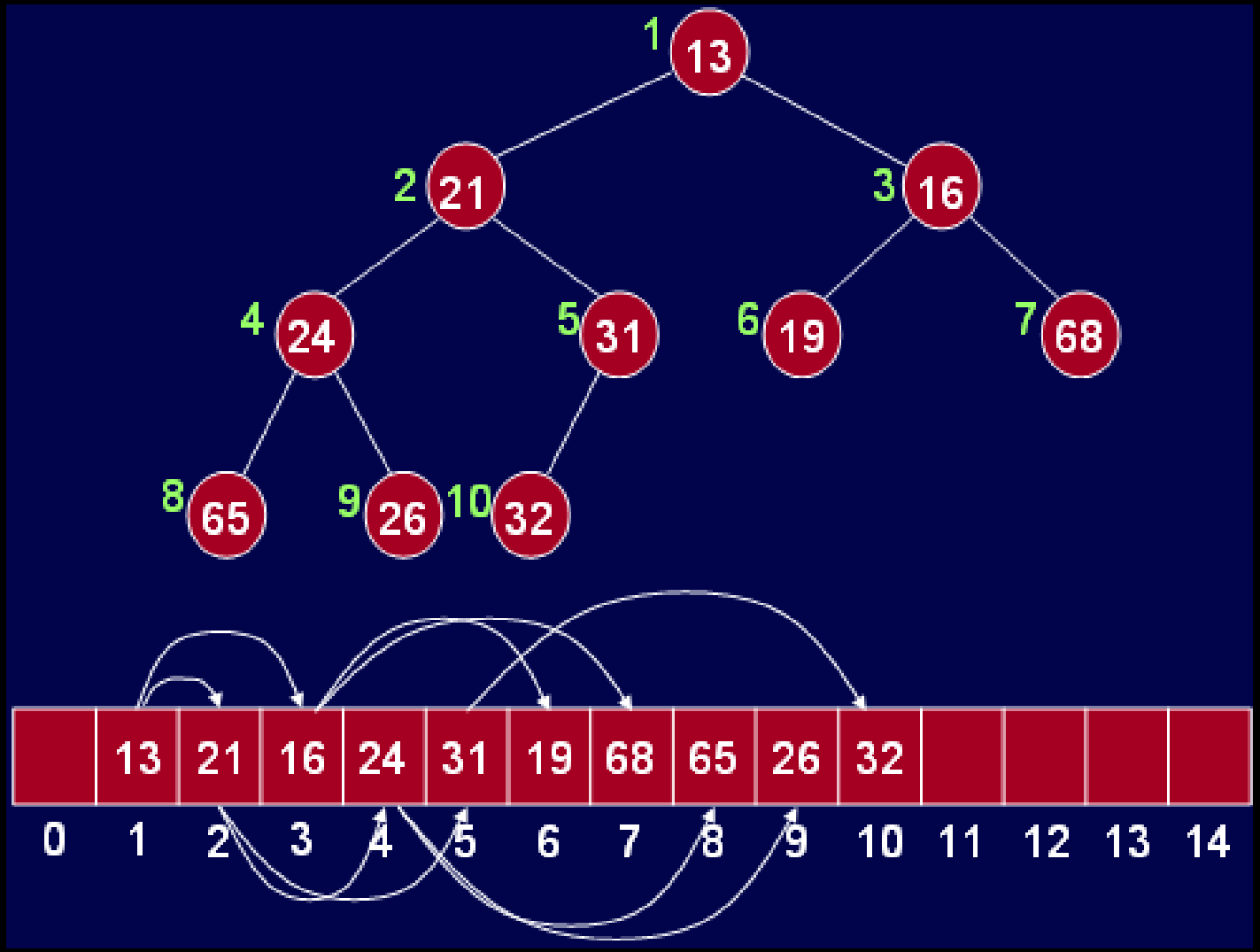
Heap

- Not a heap: heap property violated



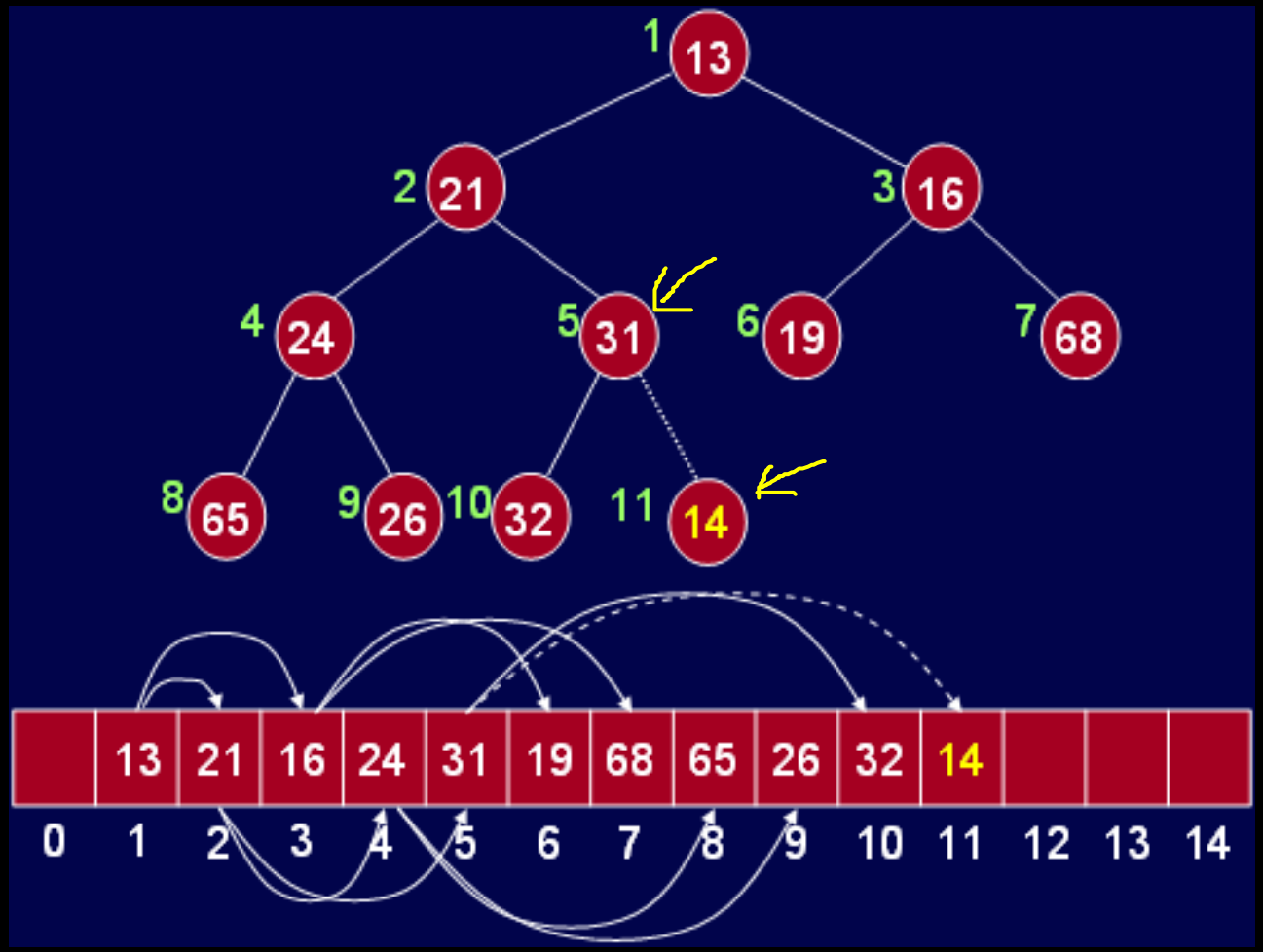
Inserting into a Heap

Assume this existing heap



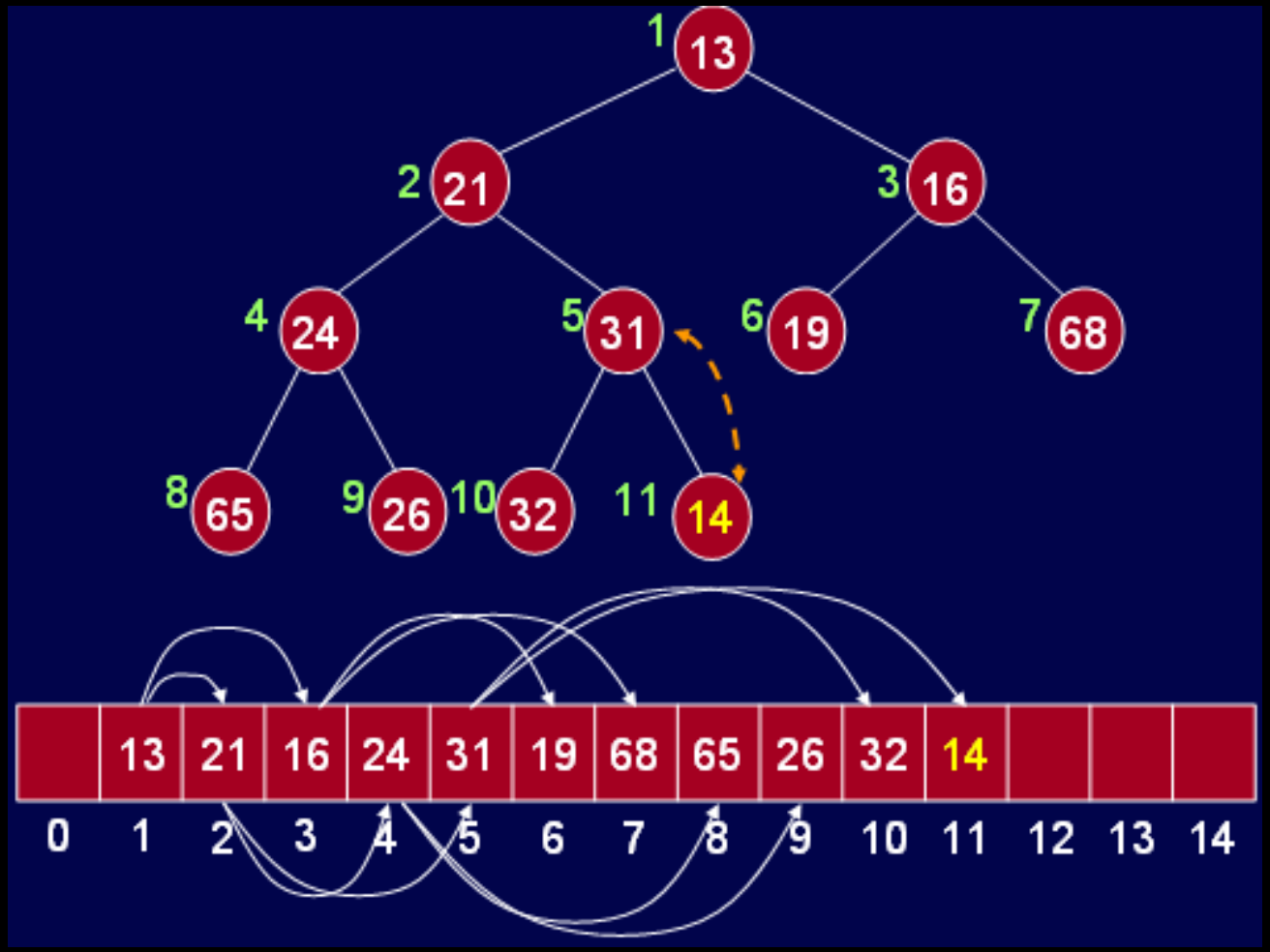
Inserting into a Heap

- insert(14)

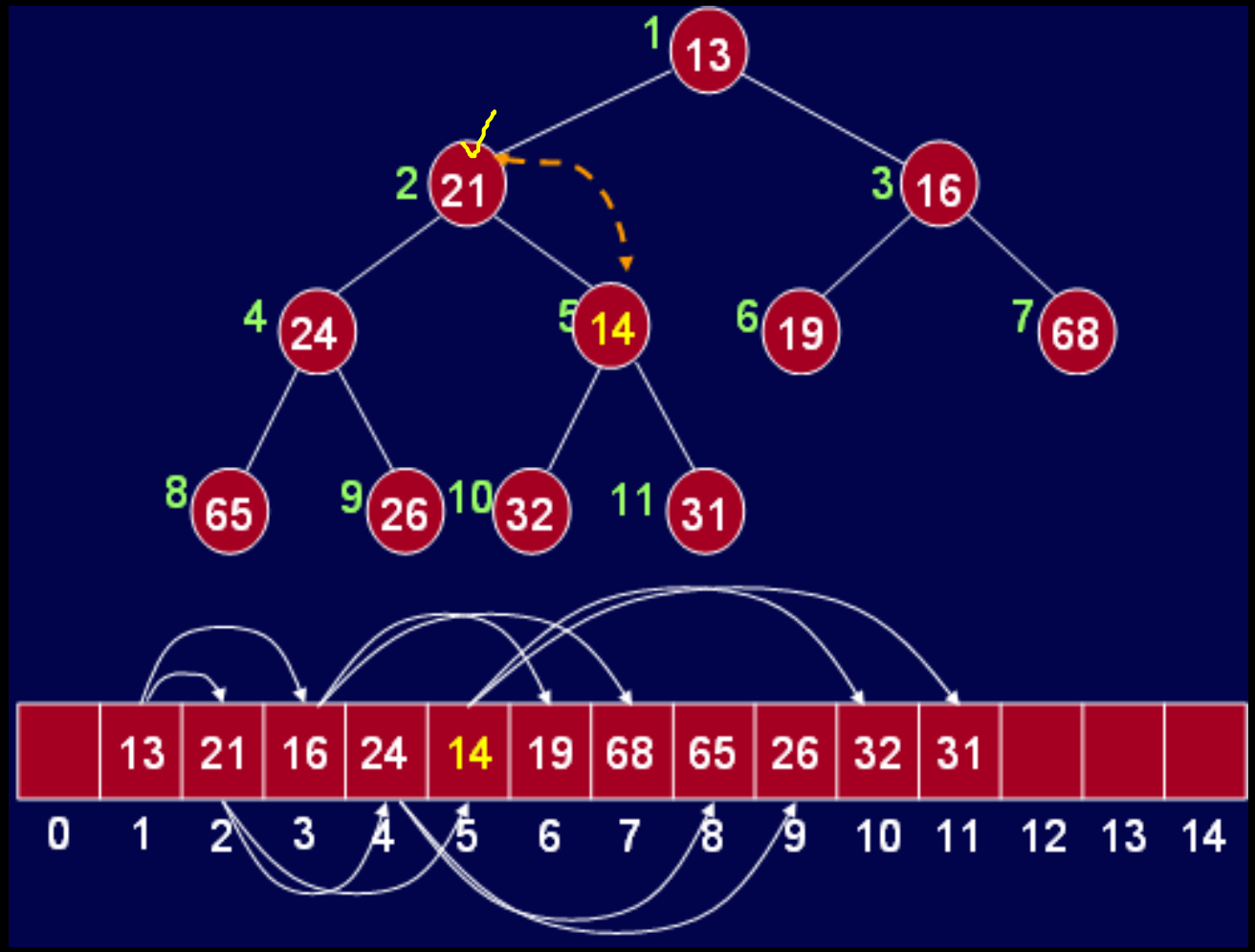


Inserting into a Heap

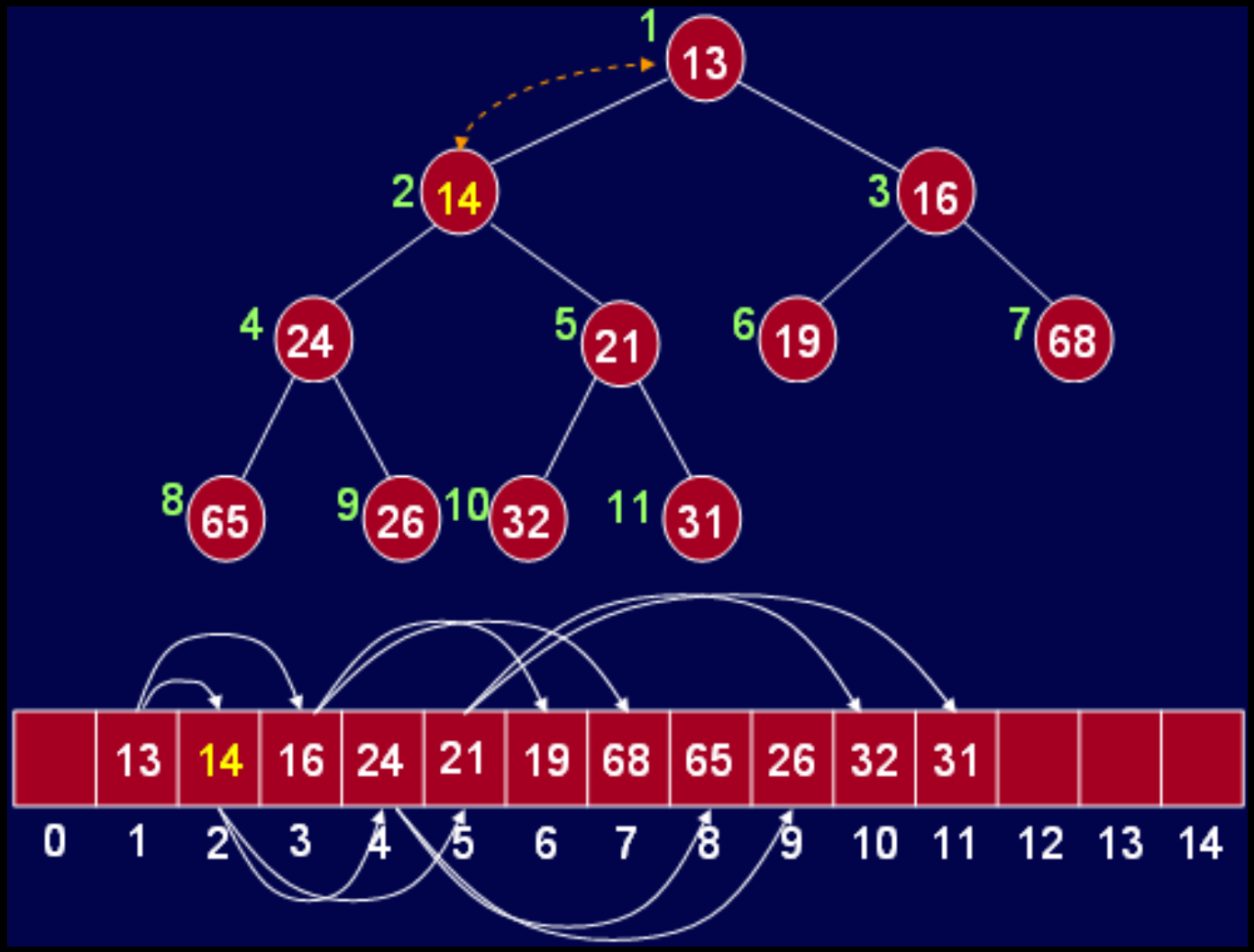
- insert(14) with exchange



Inserting into a Heap

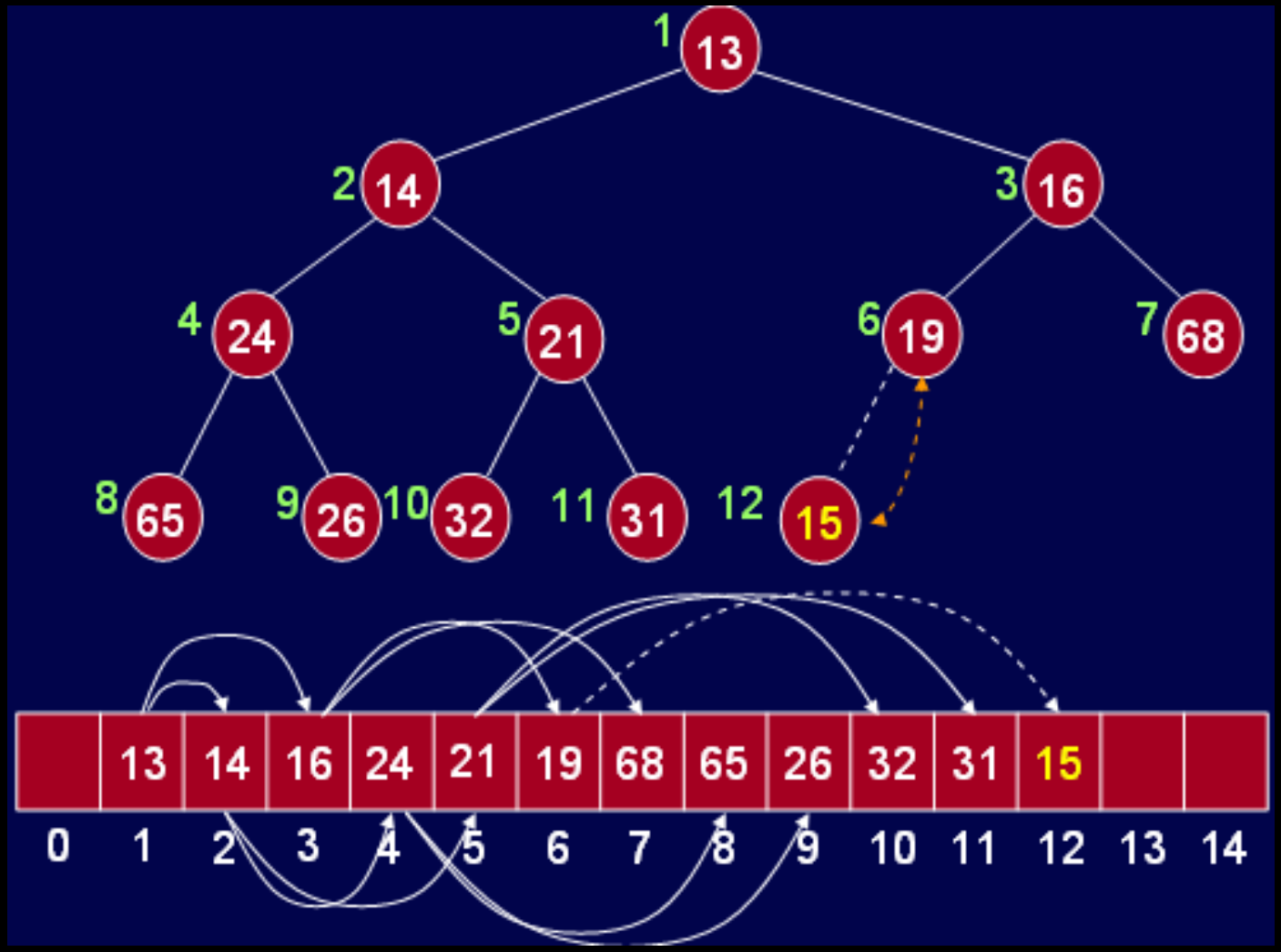


Inserting into a Heap

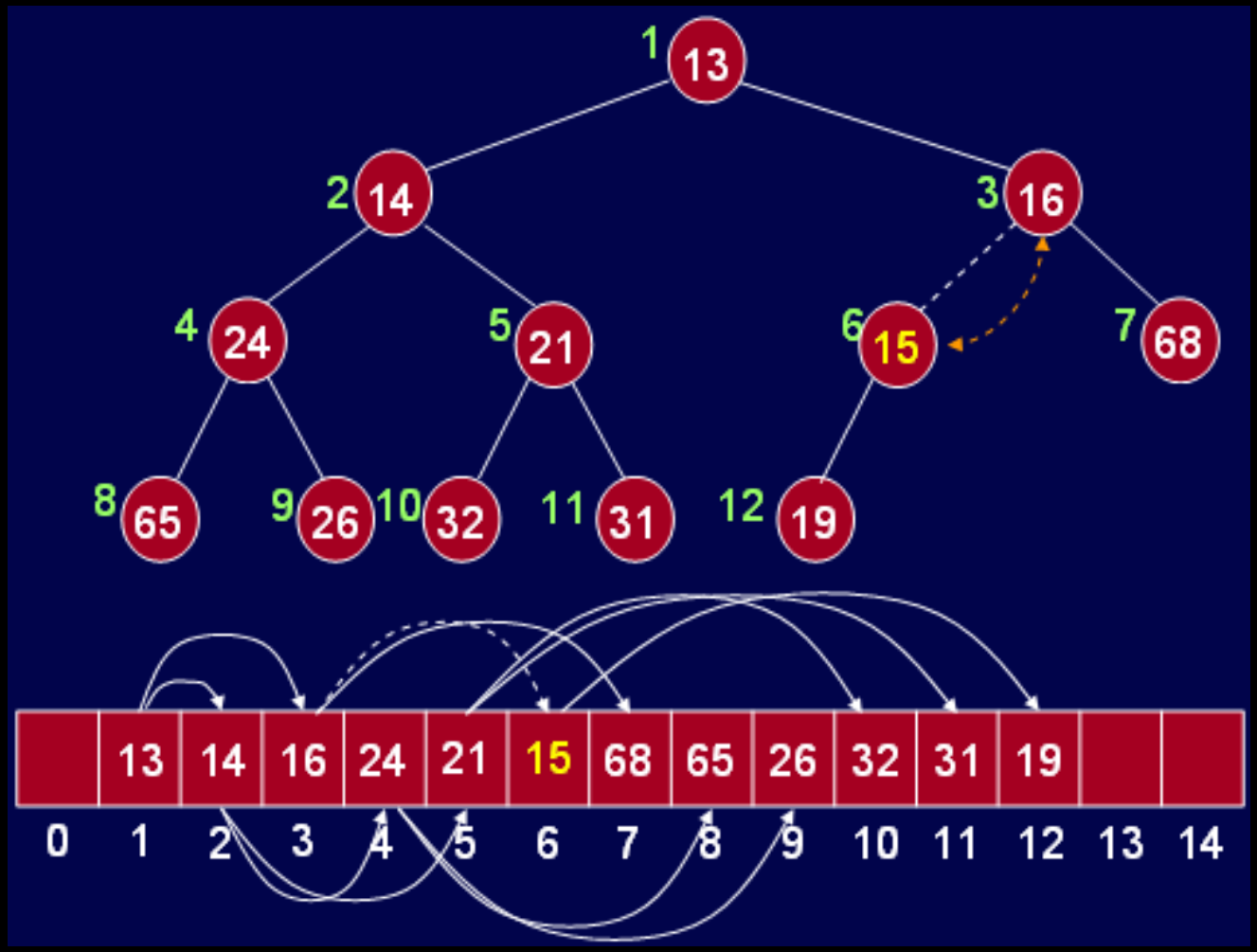


Inserting into a Heap

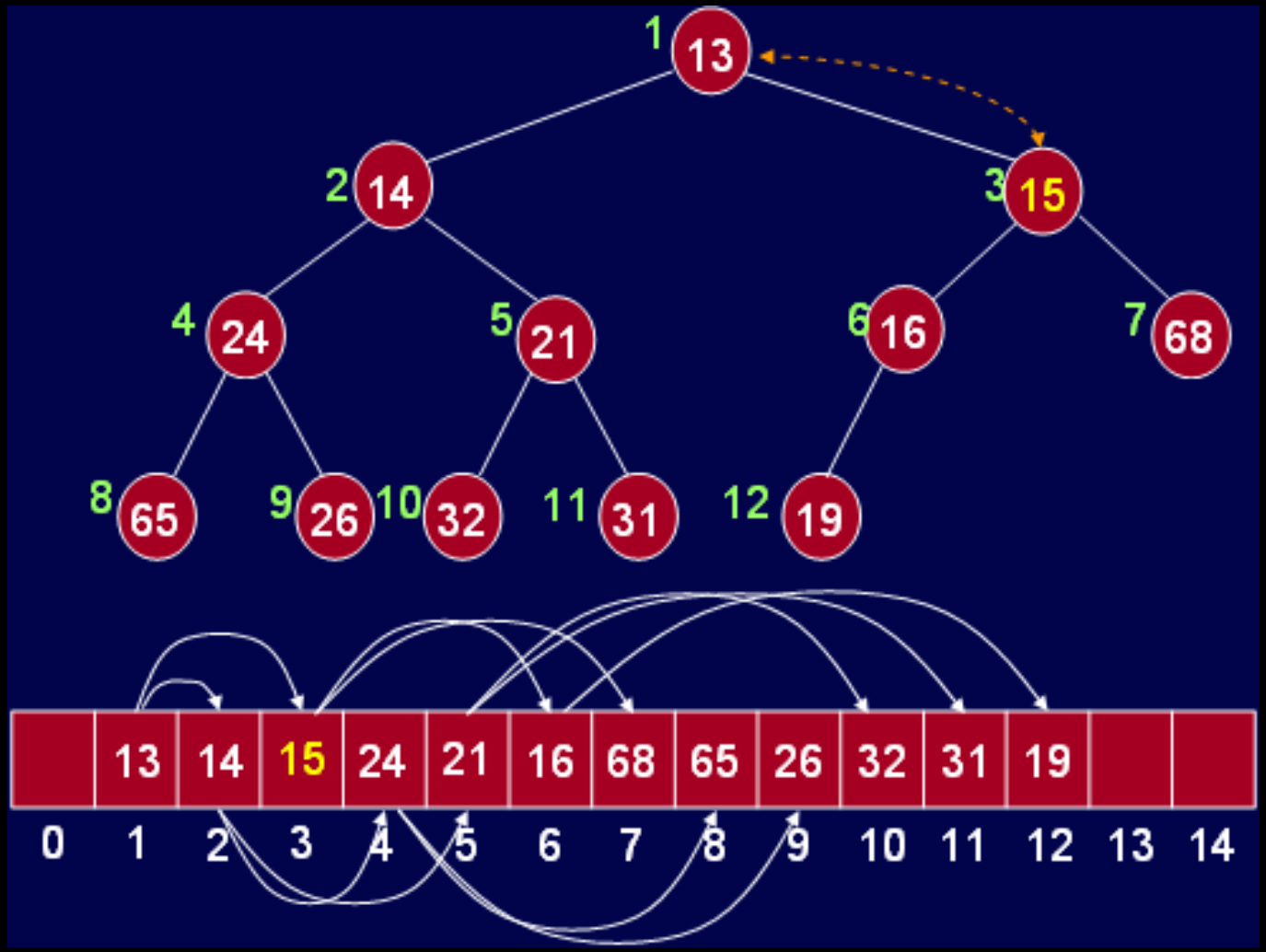
- insert(15) with exchange



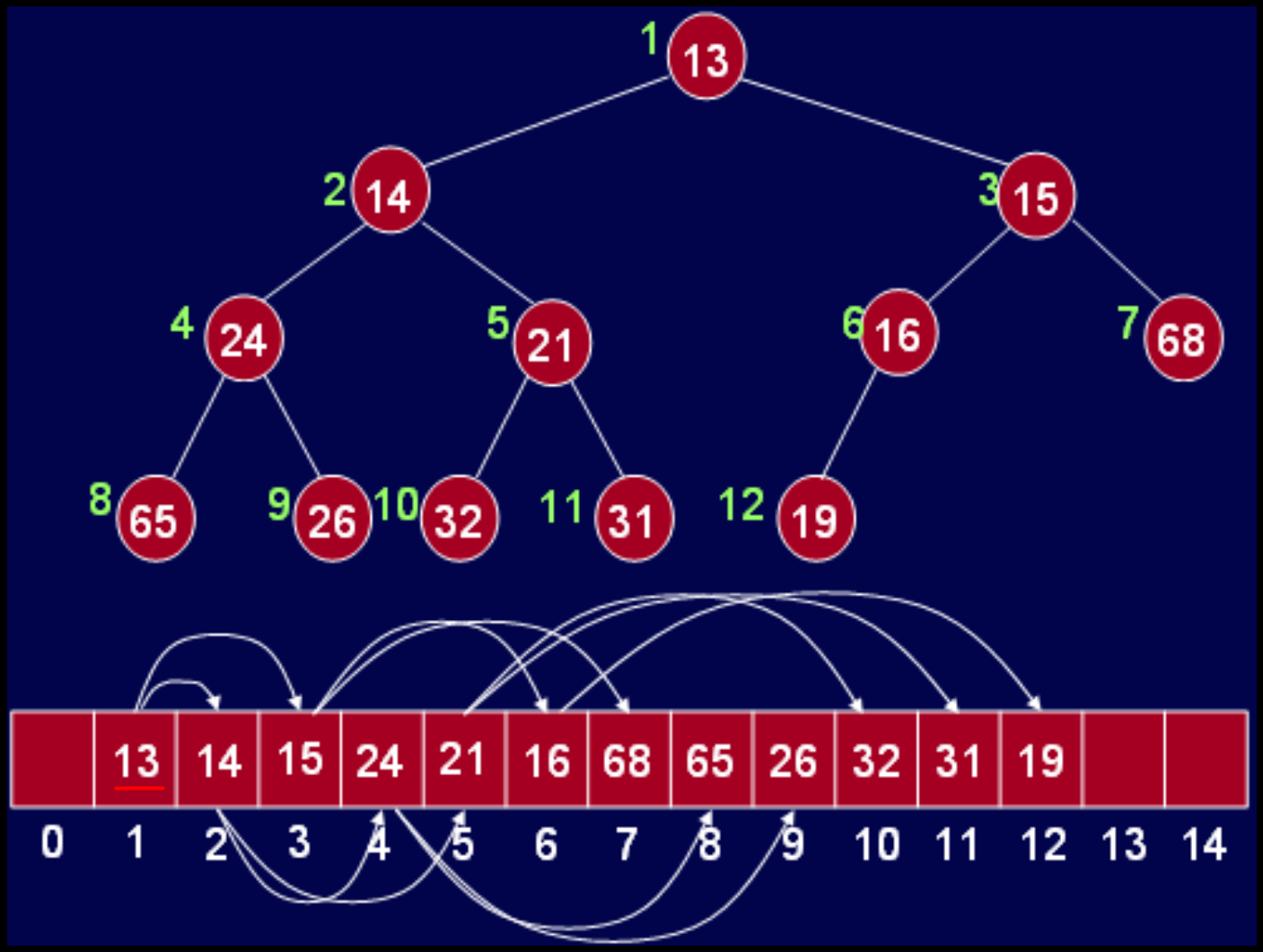
Inserting into a Heap



Inserting into a Heap



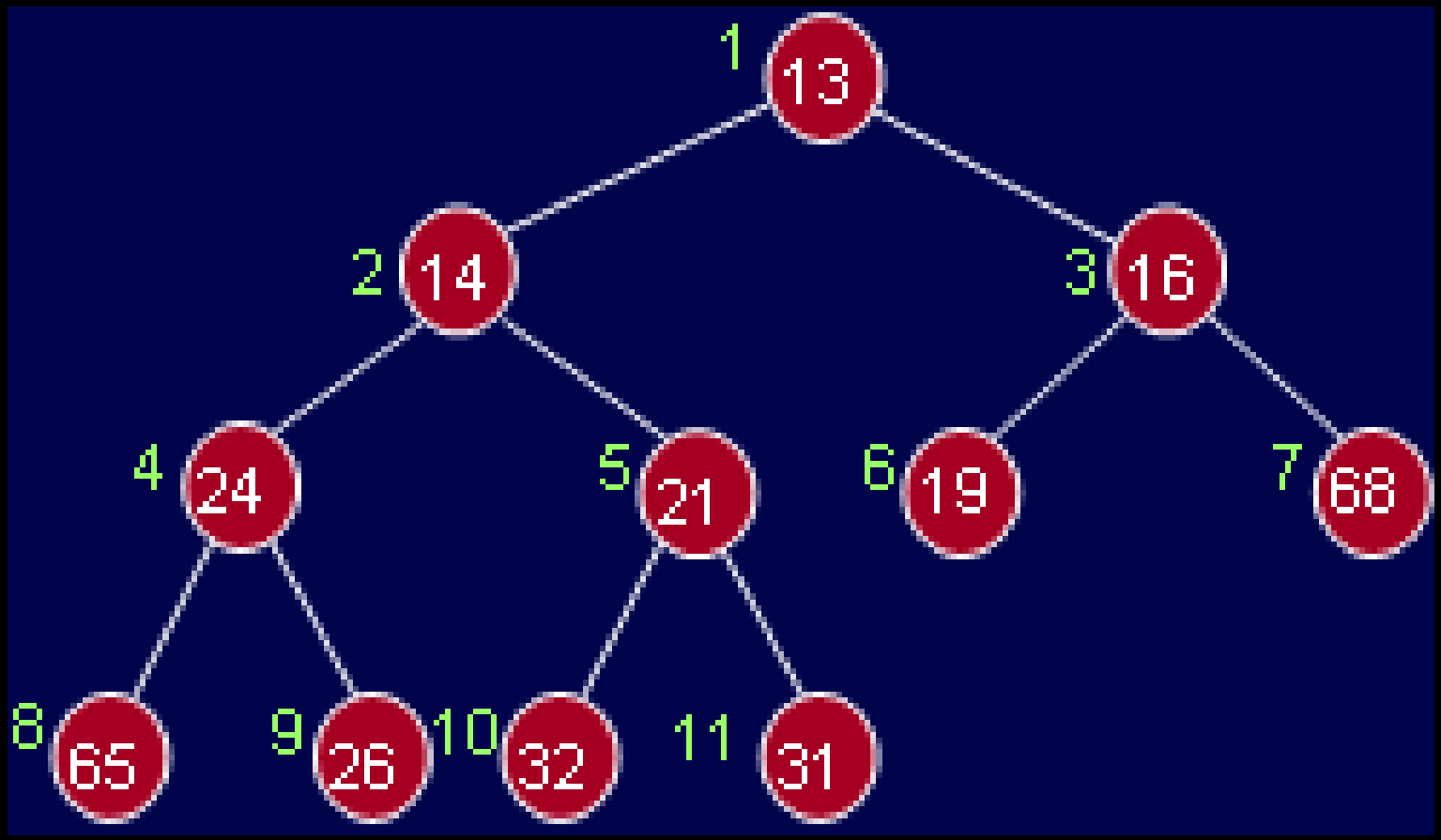
Inserting into a Heap



DeleteMin from Heap

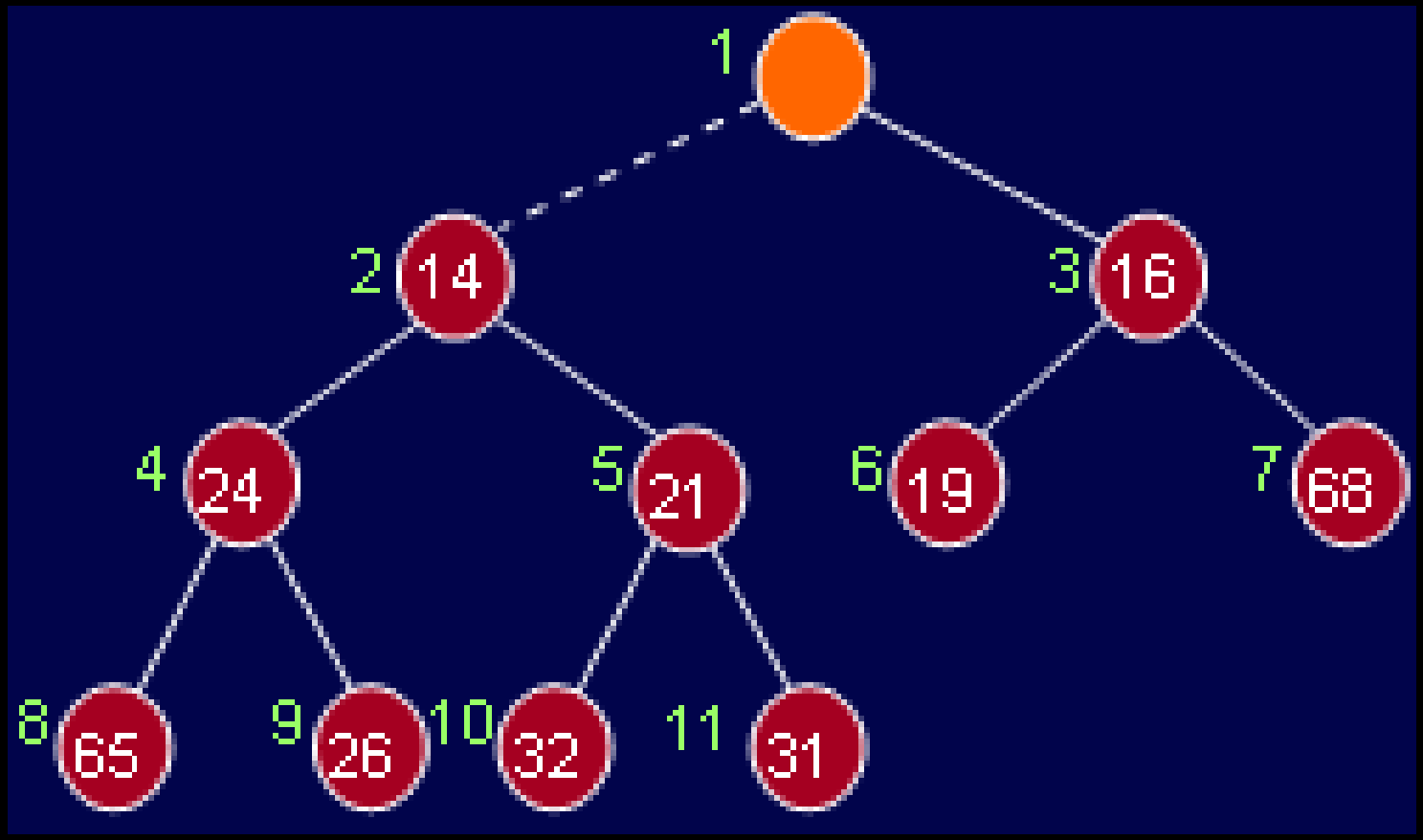
- Finding the minimum is easy; it is at the top of the heap.
- Deleting it (or removing it) causes a hole which needs to be filled.

DeleteMin from Heap



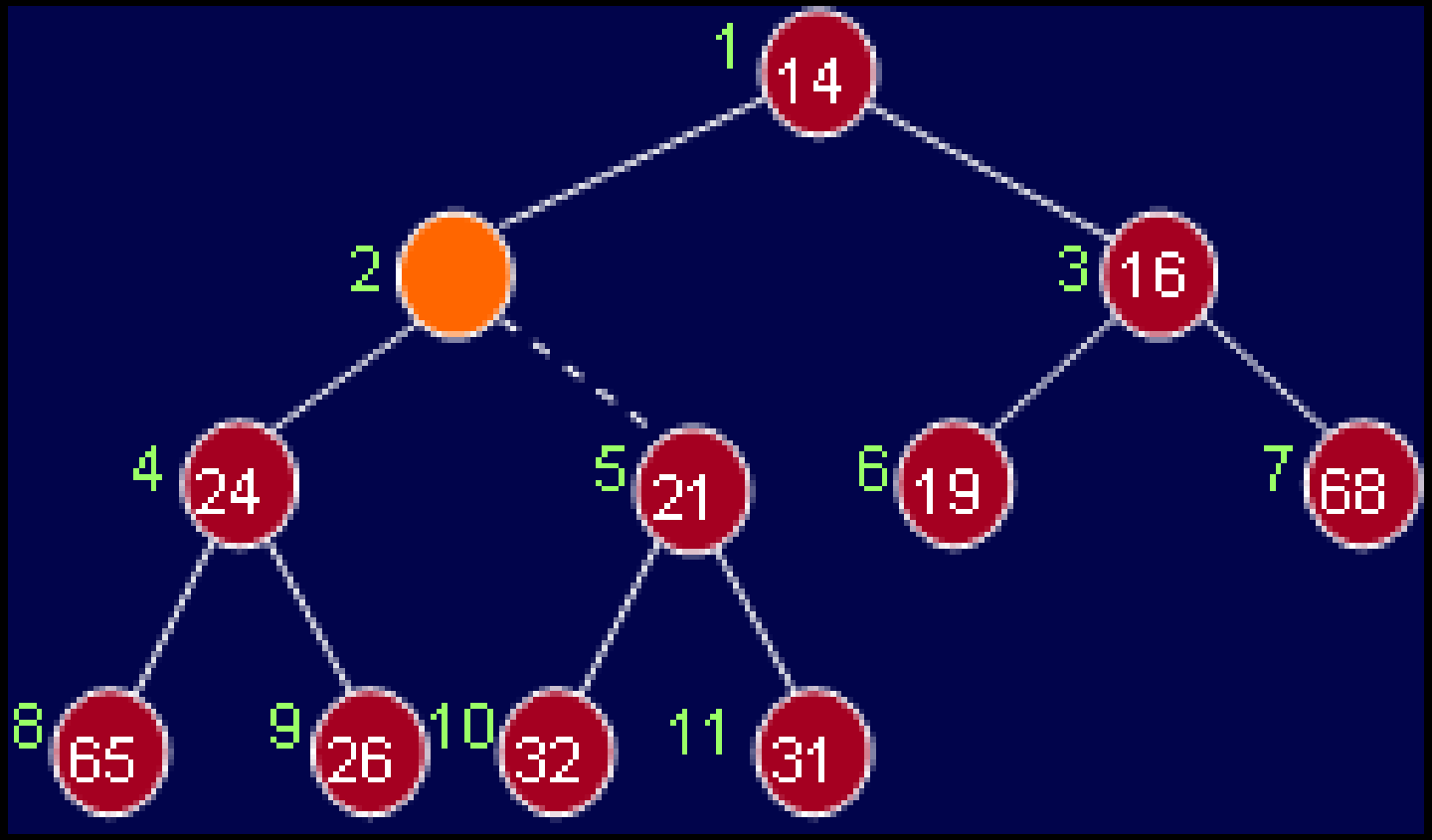
DeleteMin from Heap

- deleteMin()



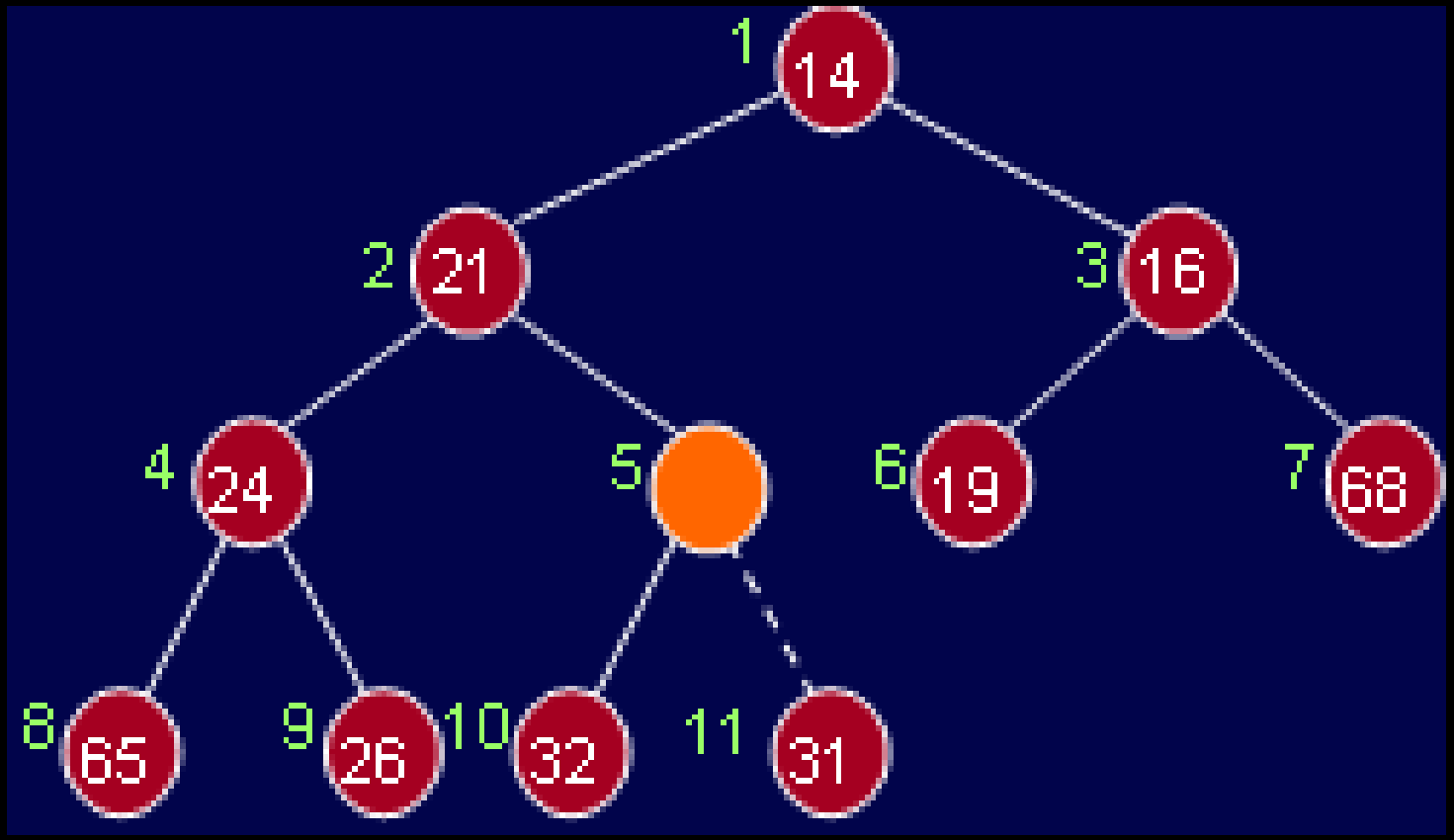
DeleteMin from Heap

- deleteMin()



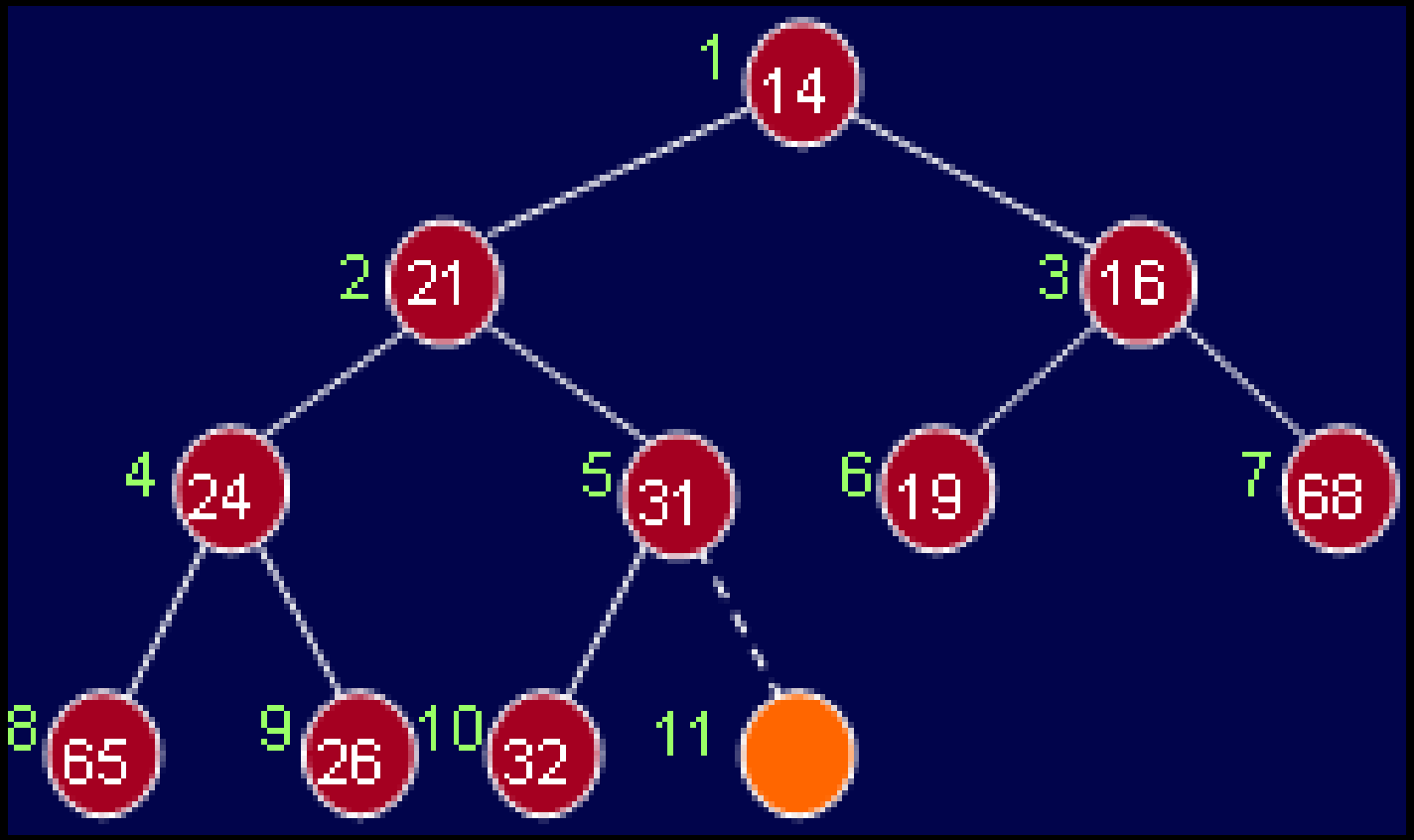
DeleteMin from Heap

- deleteMin()



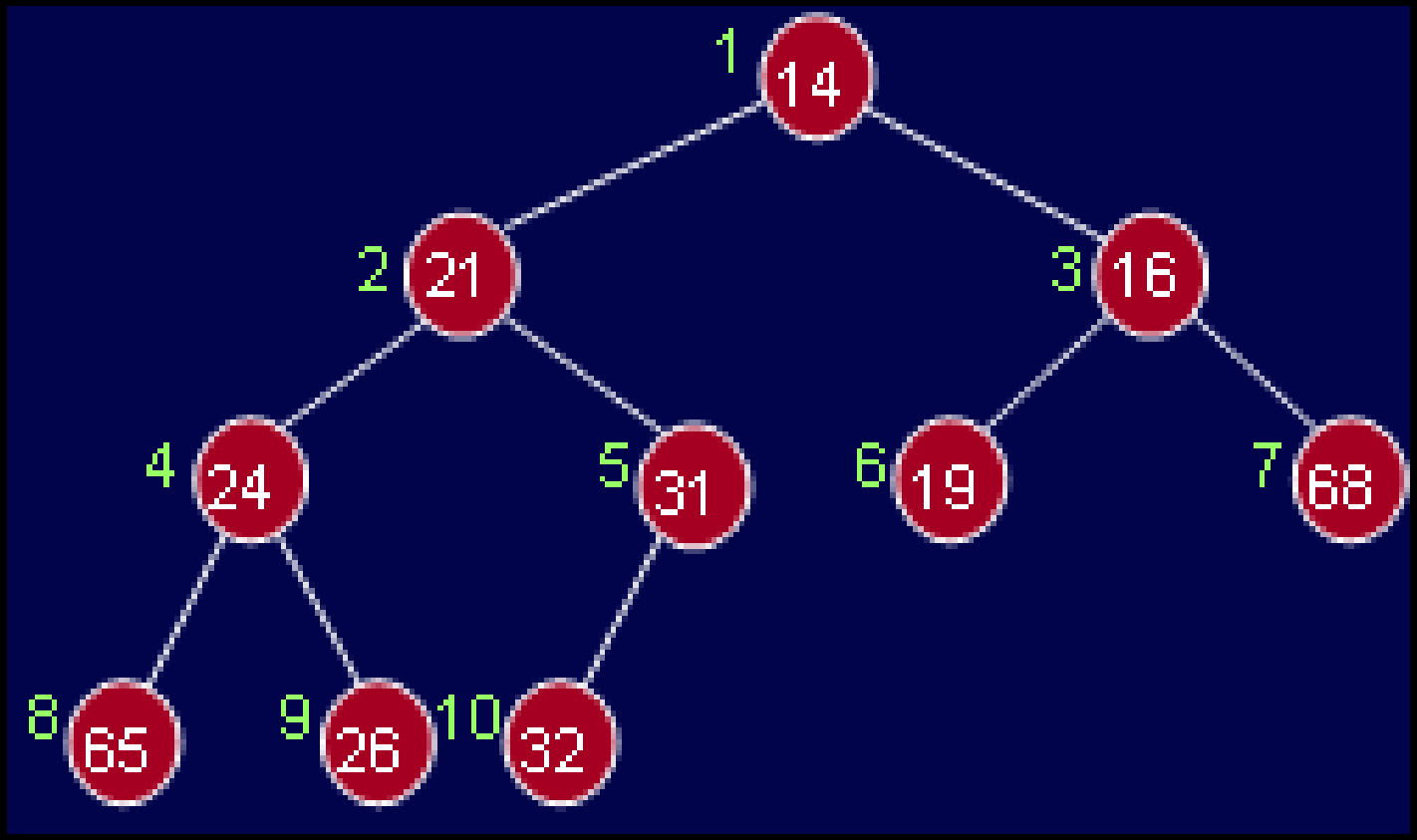
DeleteMin from Heap

- deleteMin()



DeleteMin from Heap

- deleteMin(): heap size is reduced by 1.



Thank You …