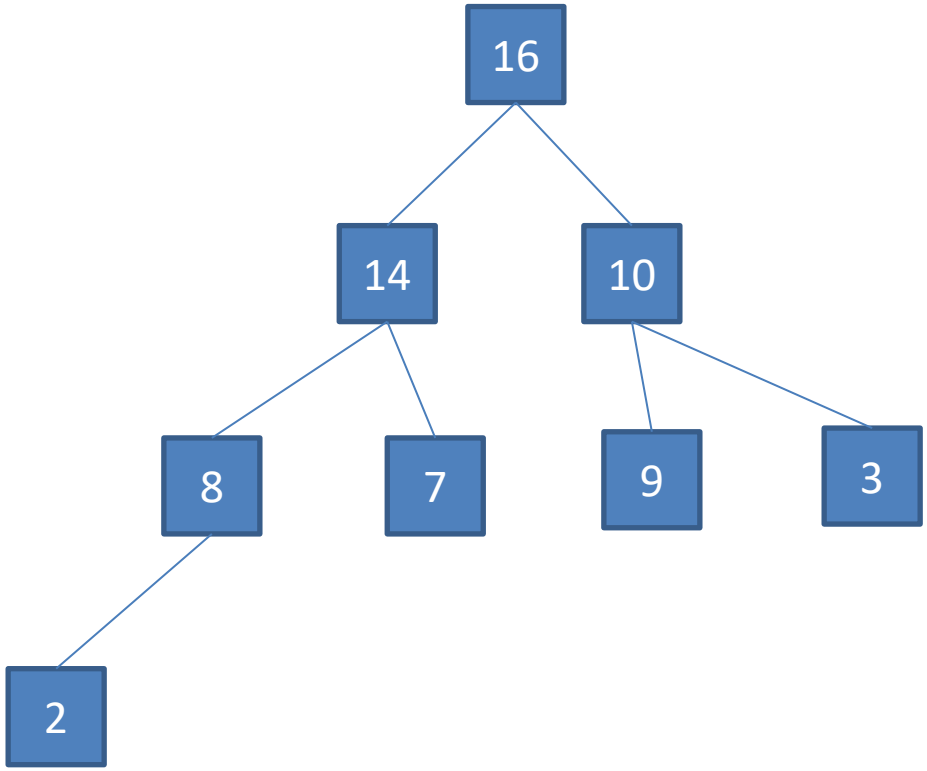
Heapsort Example



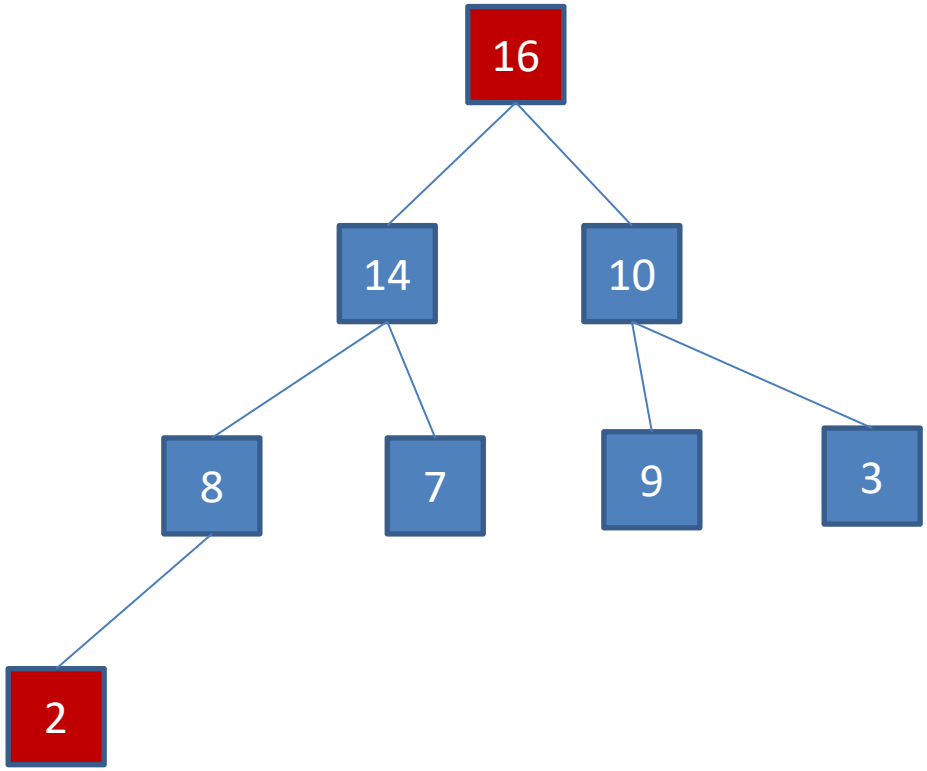
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



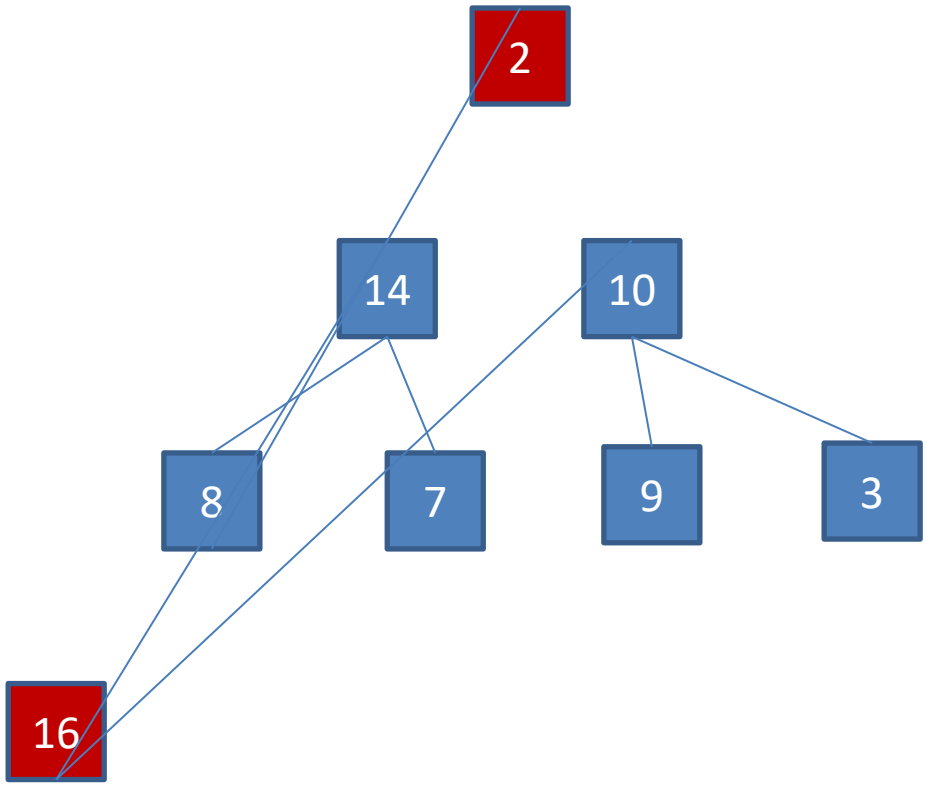
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



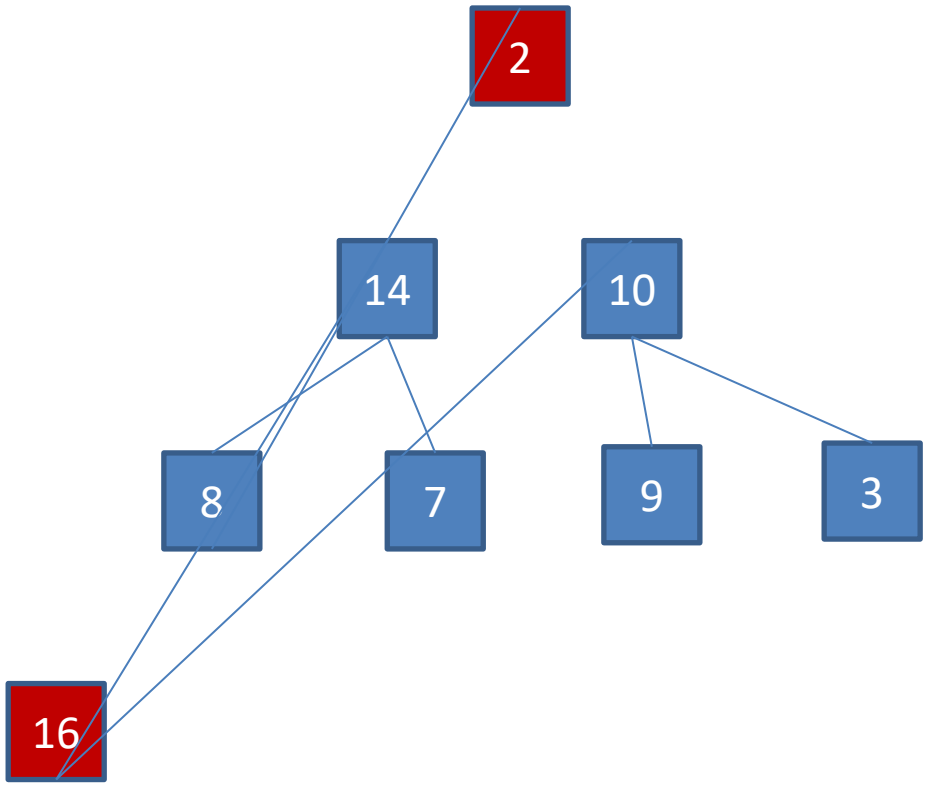
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



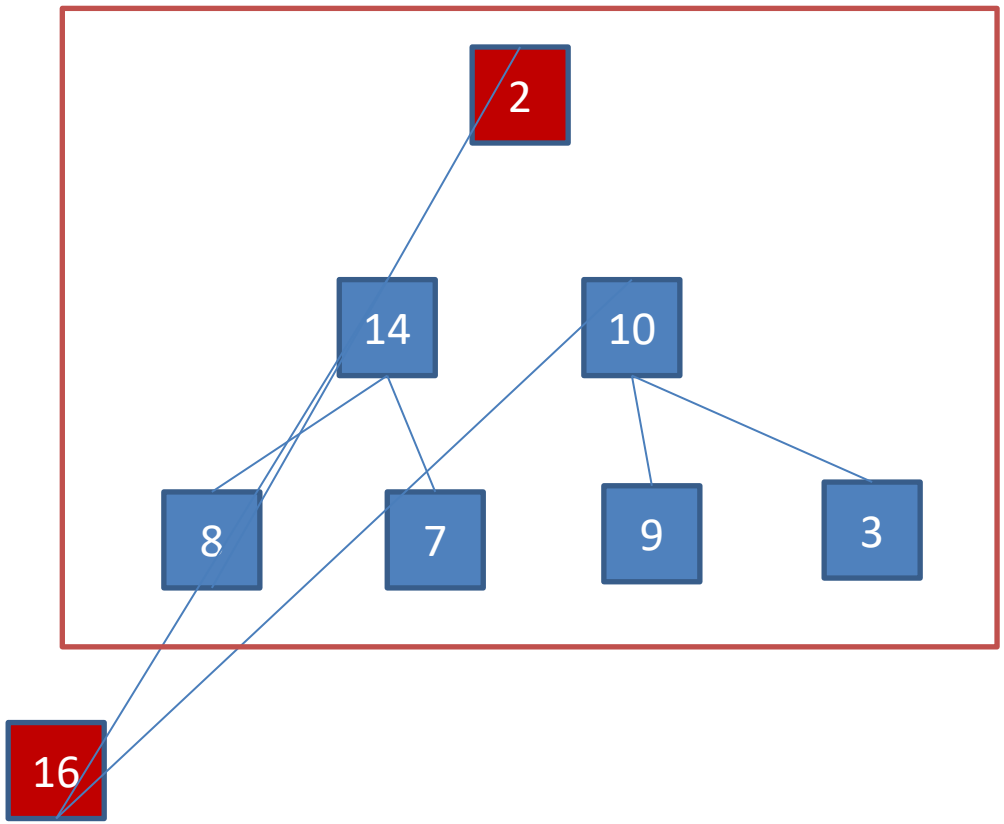
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



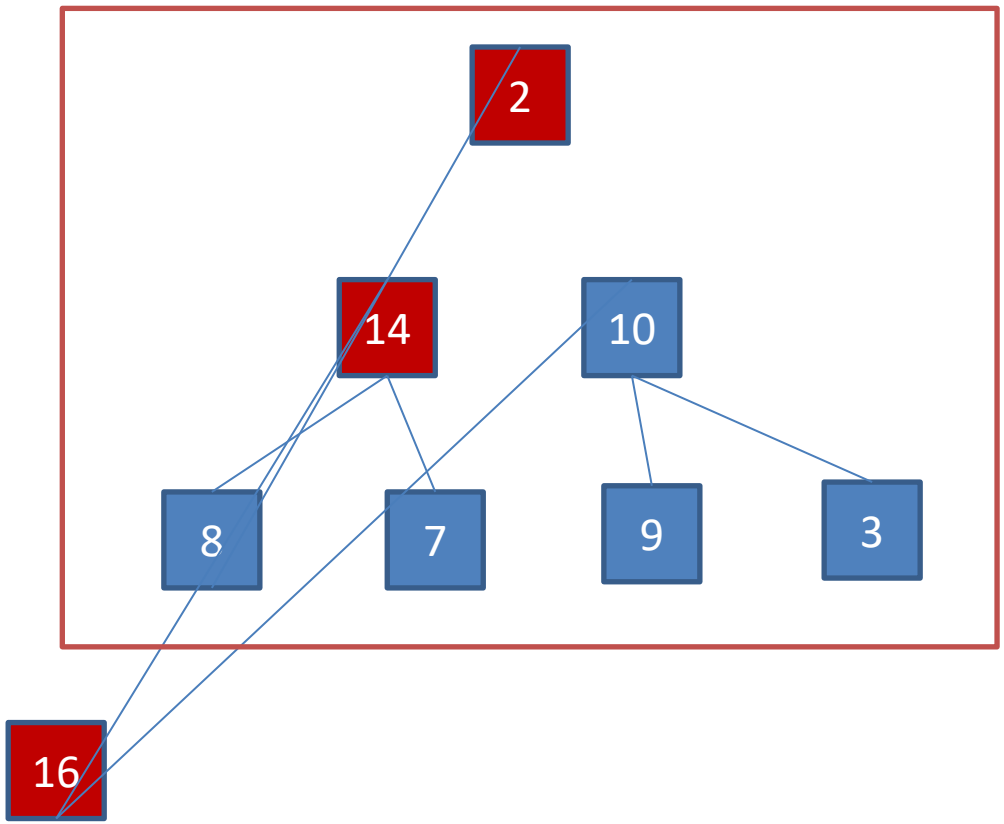
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



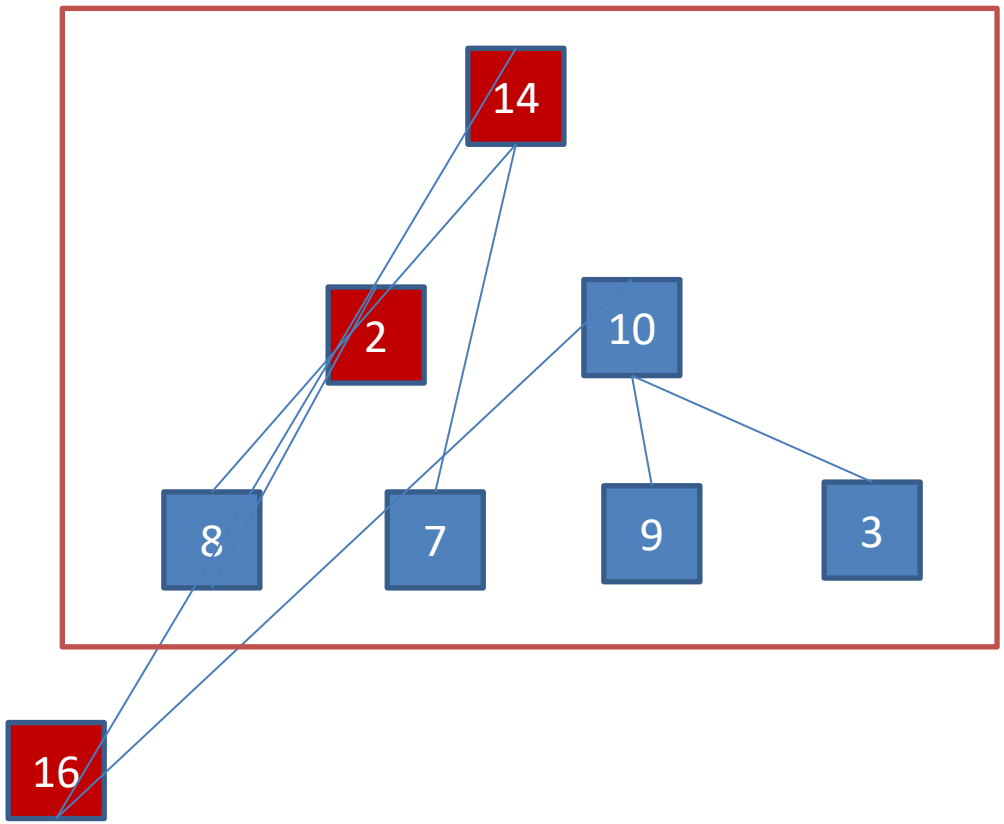
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



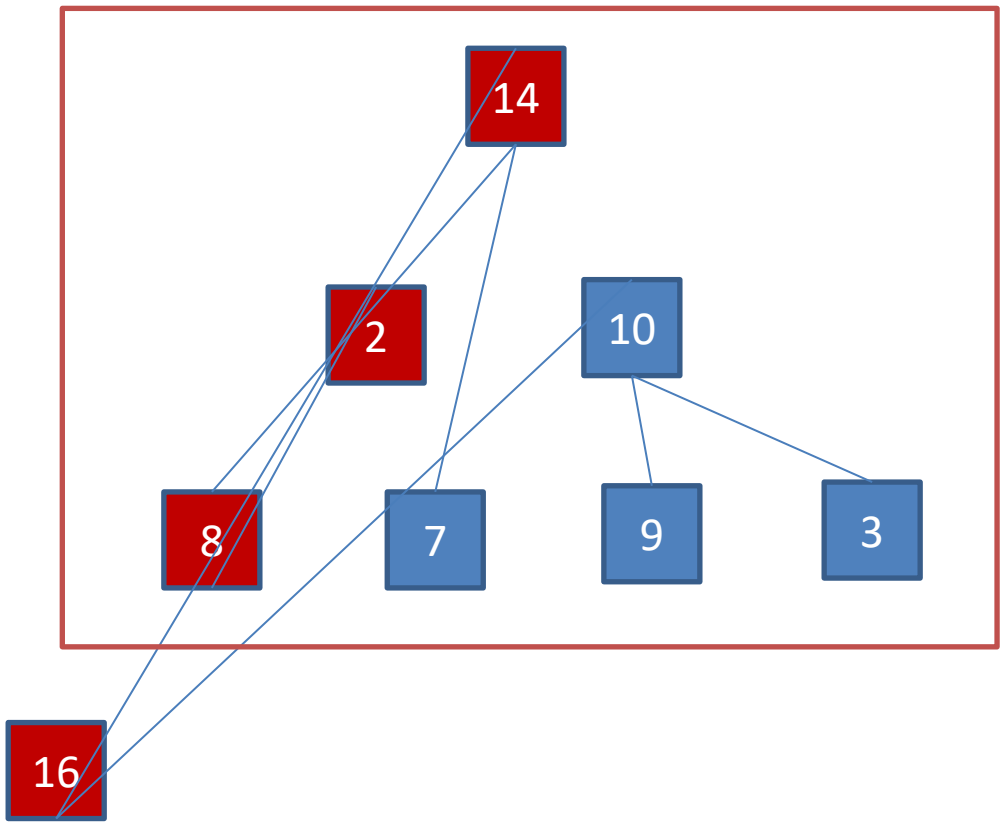
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



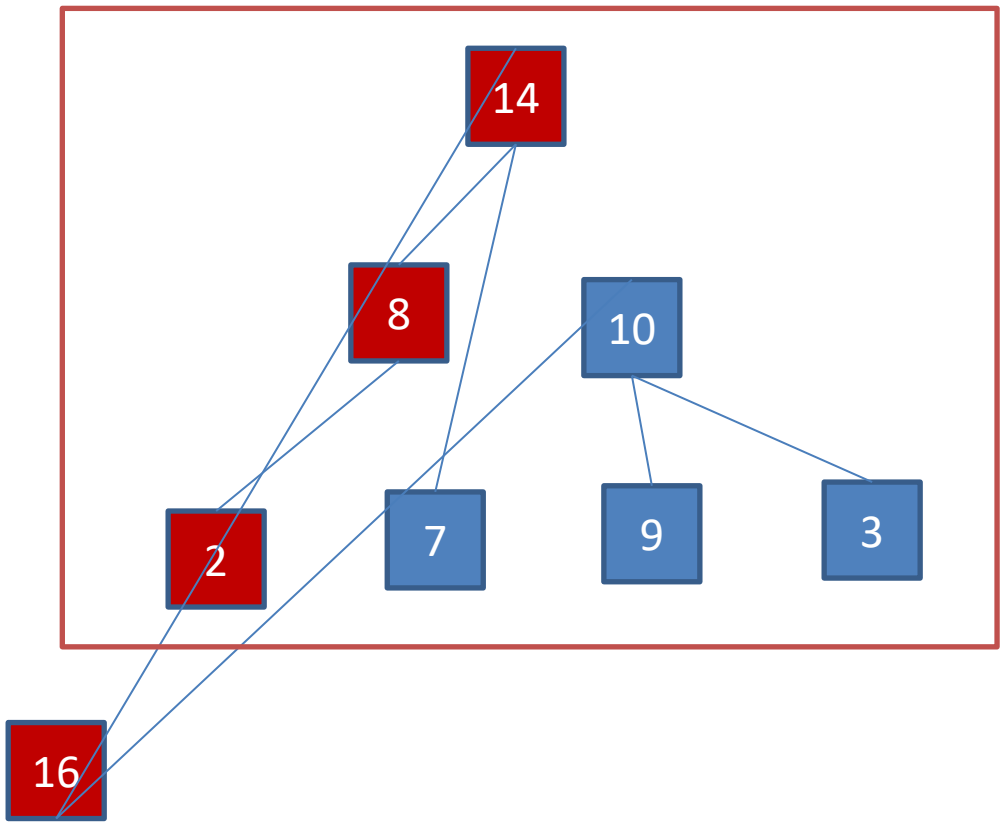
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



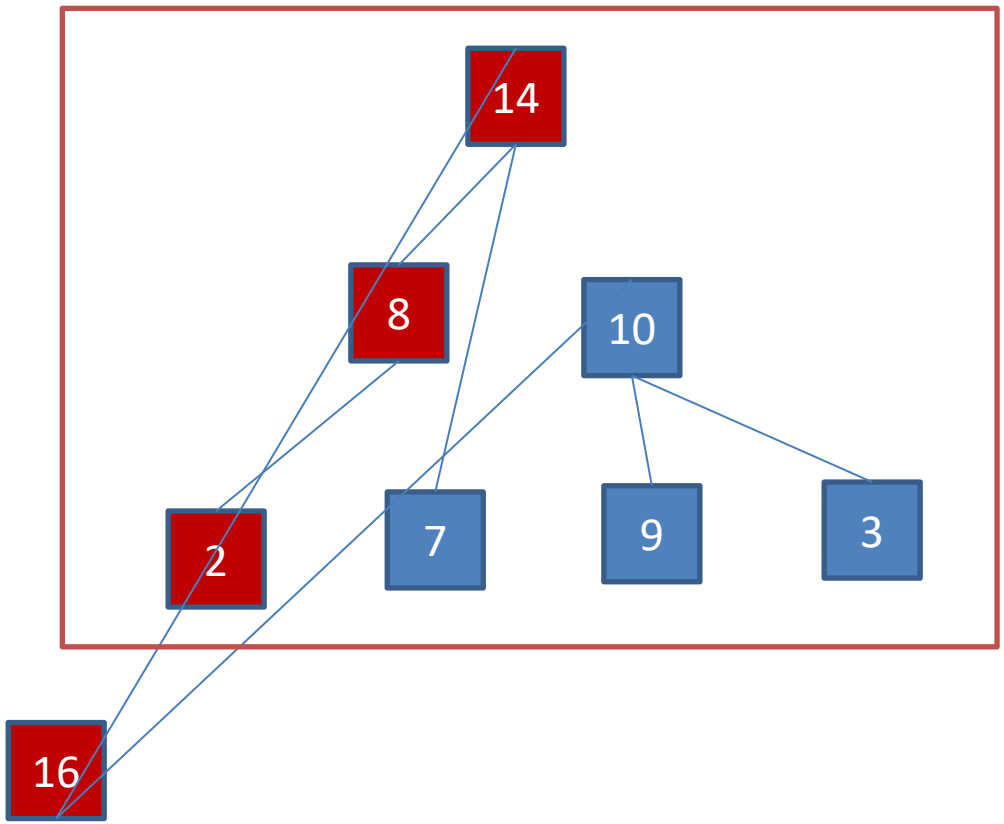
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



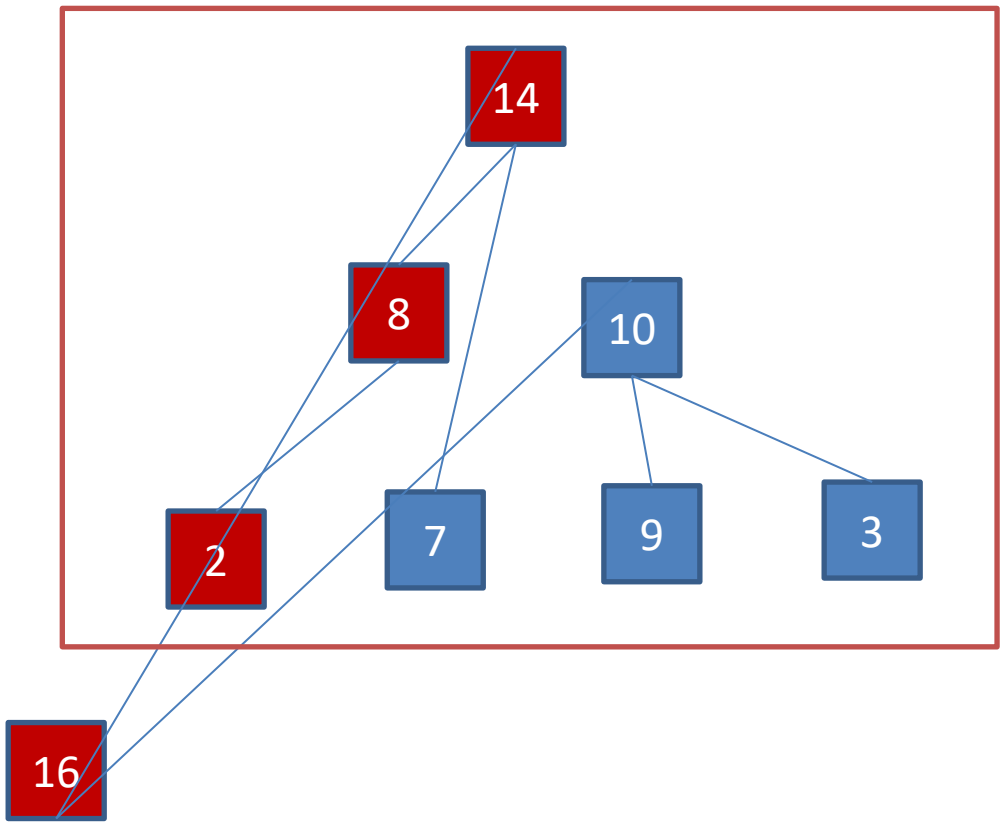
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



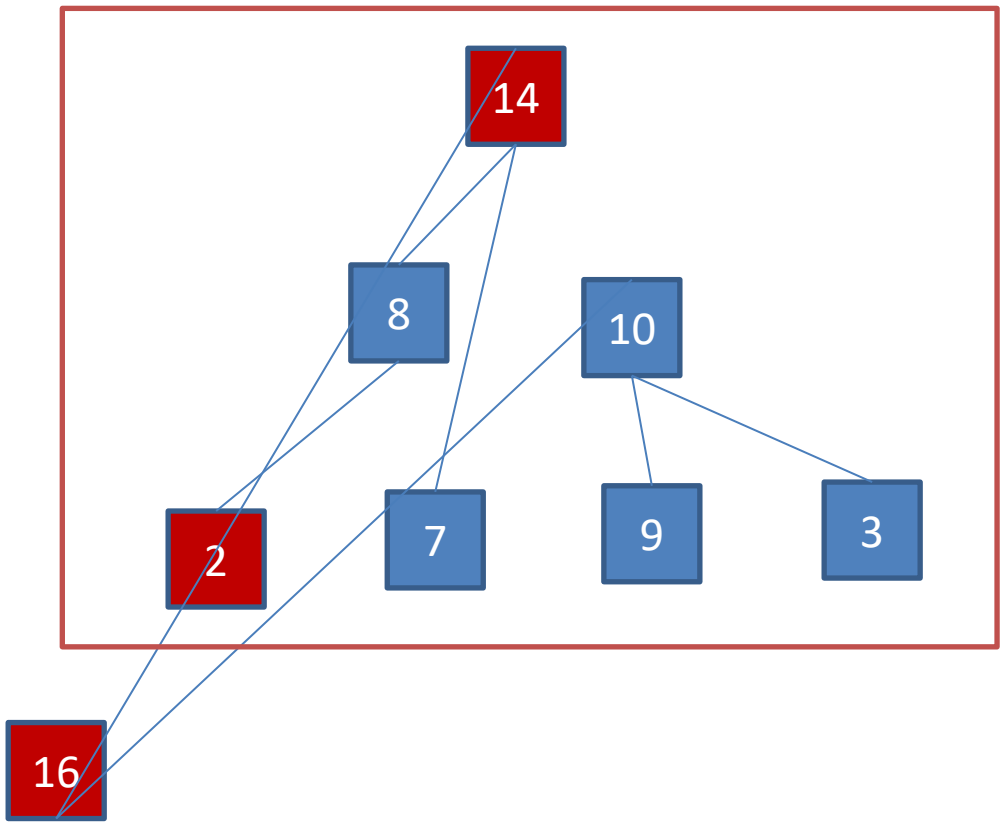
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



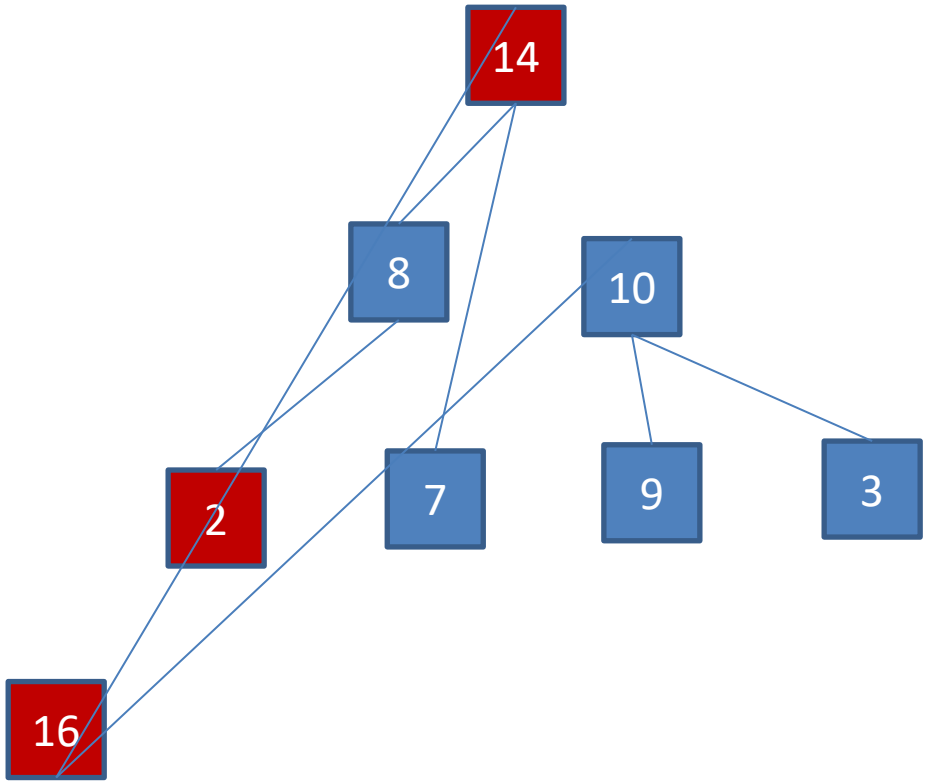
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



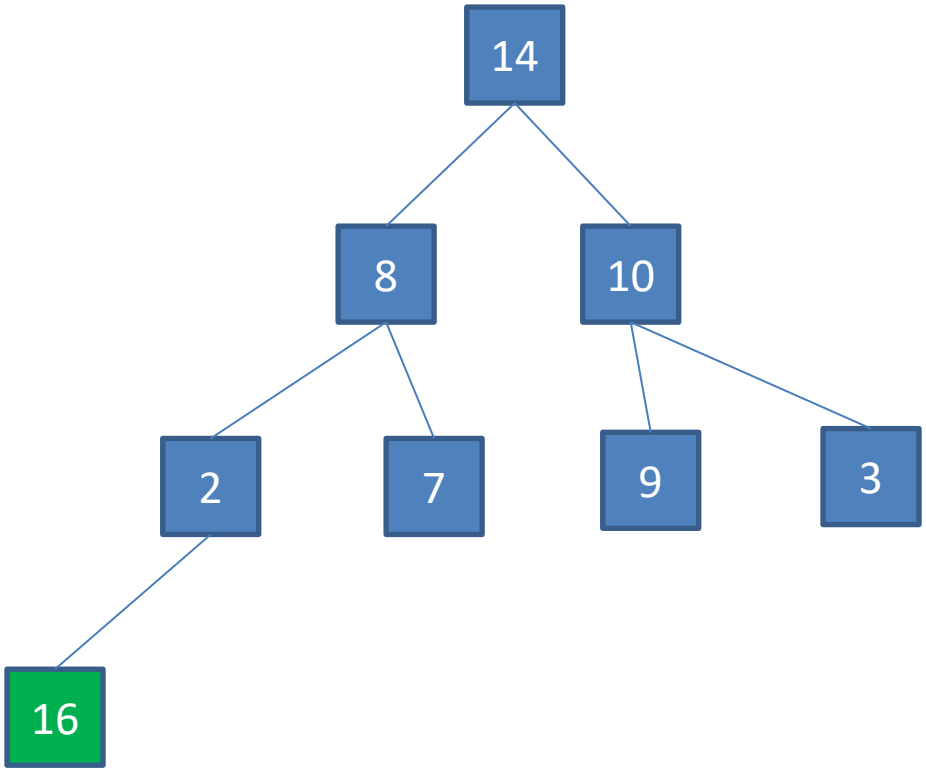
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



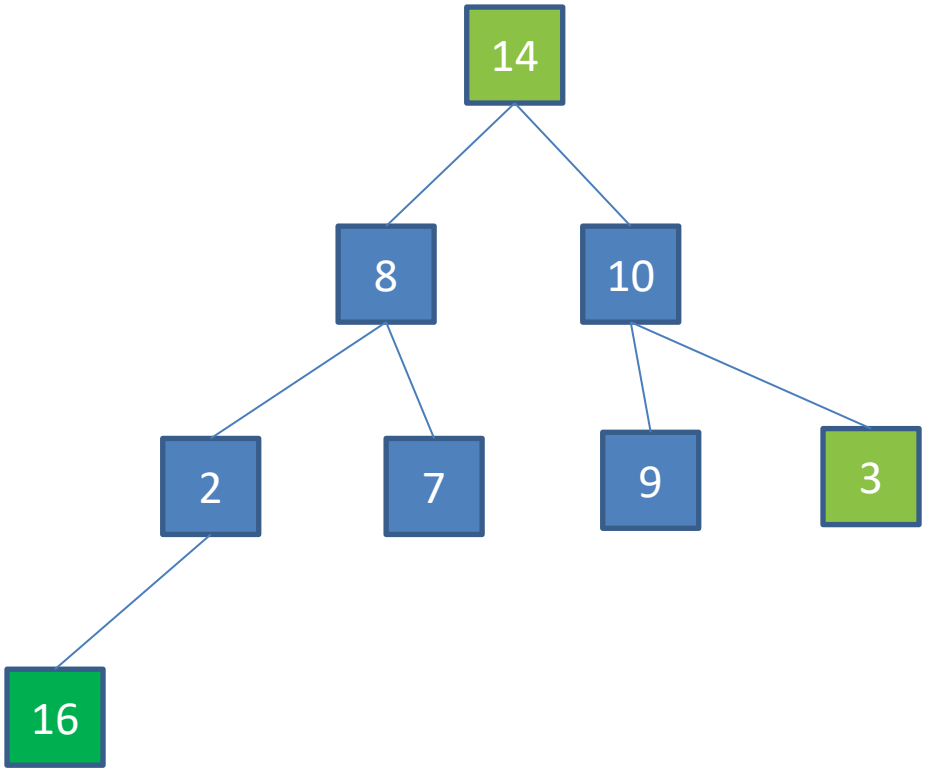
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



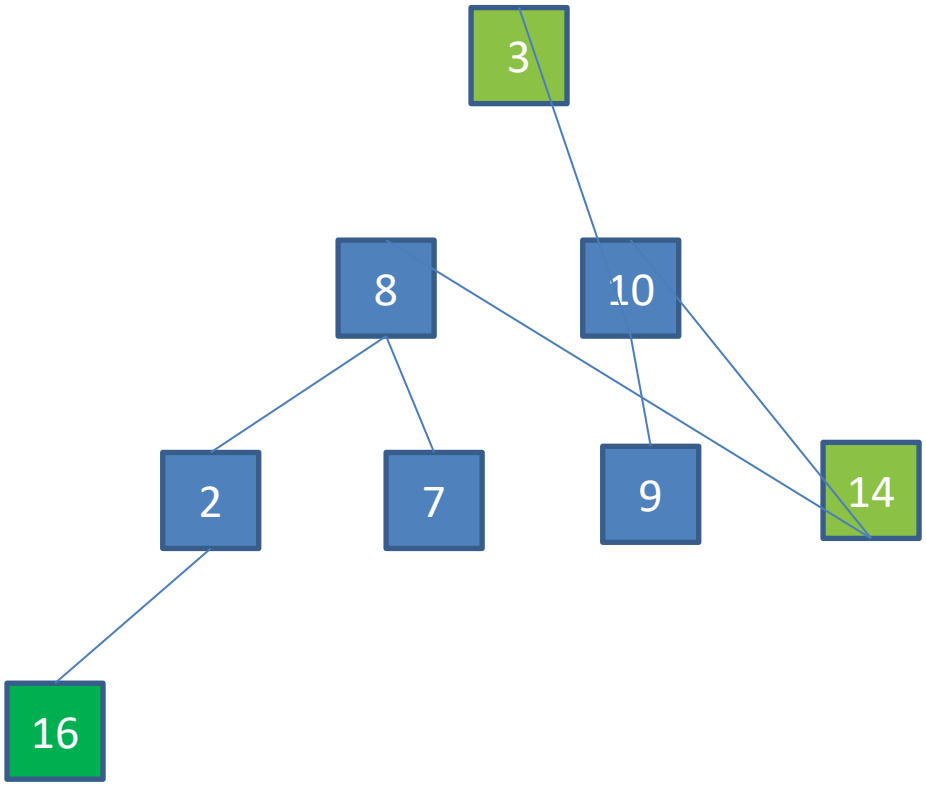
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



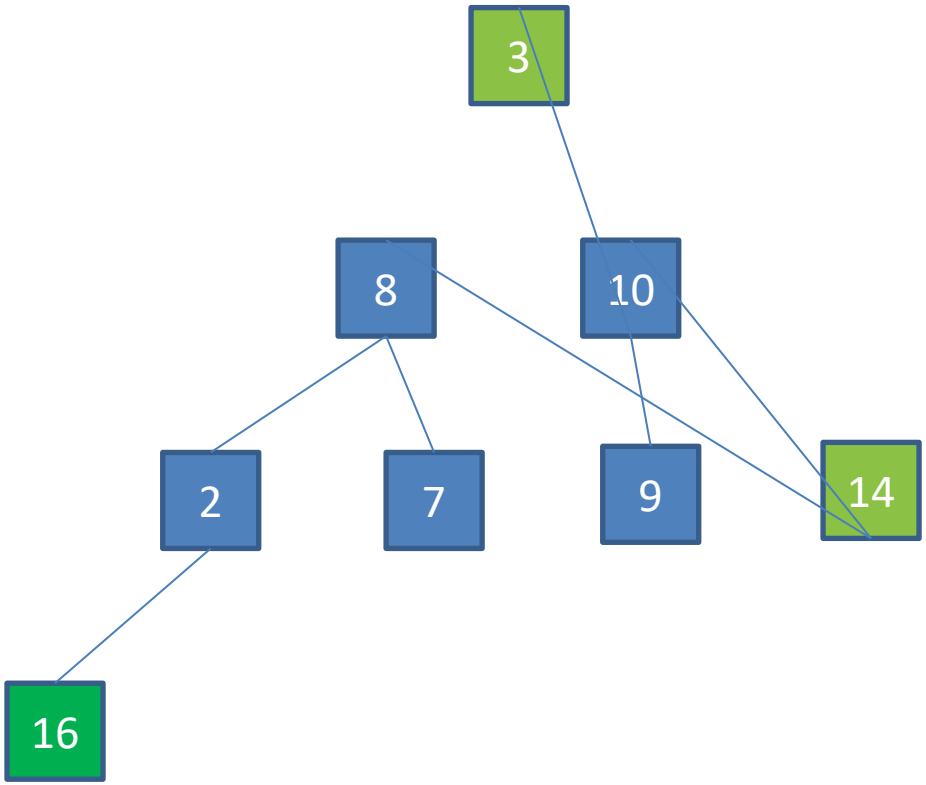
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



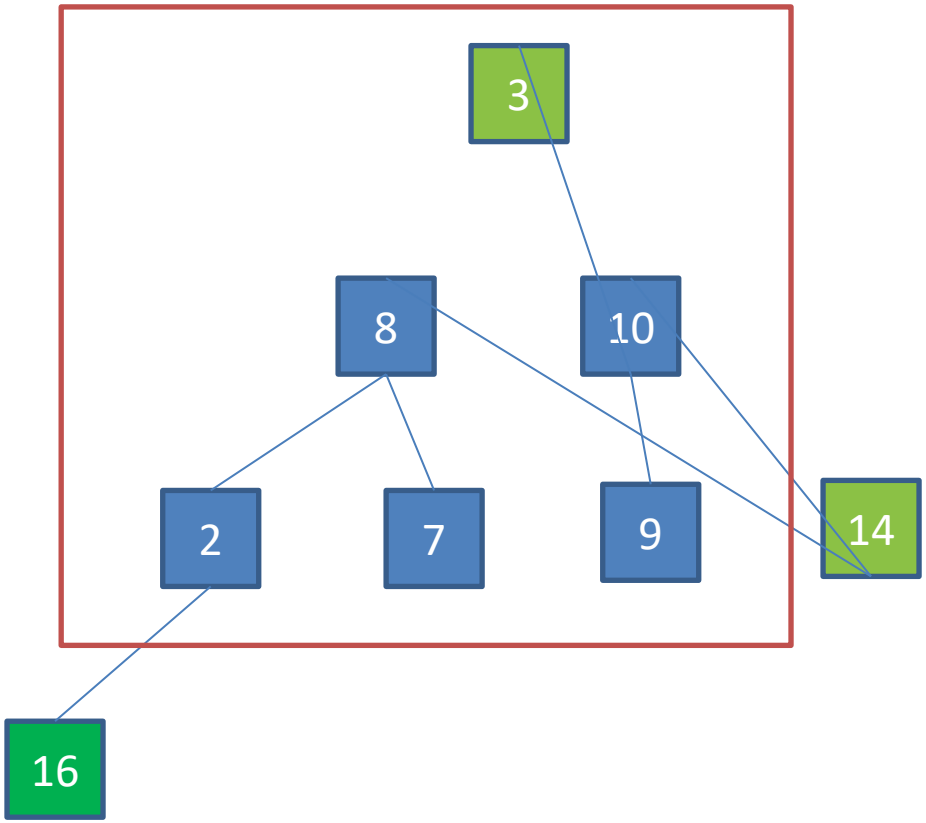
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



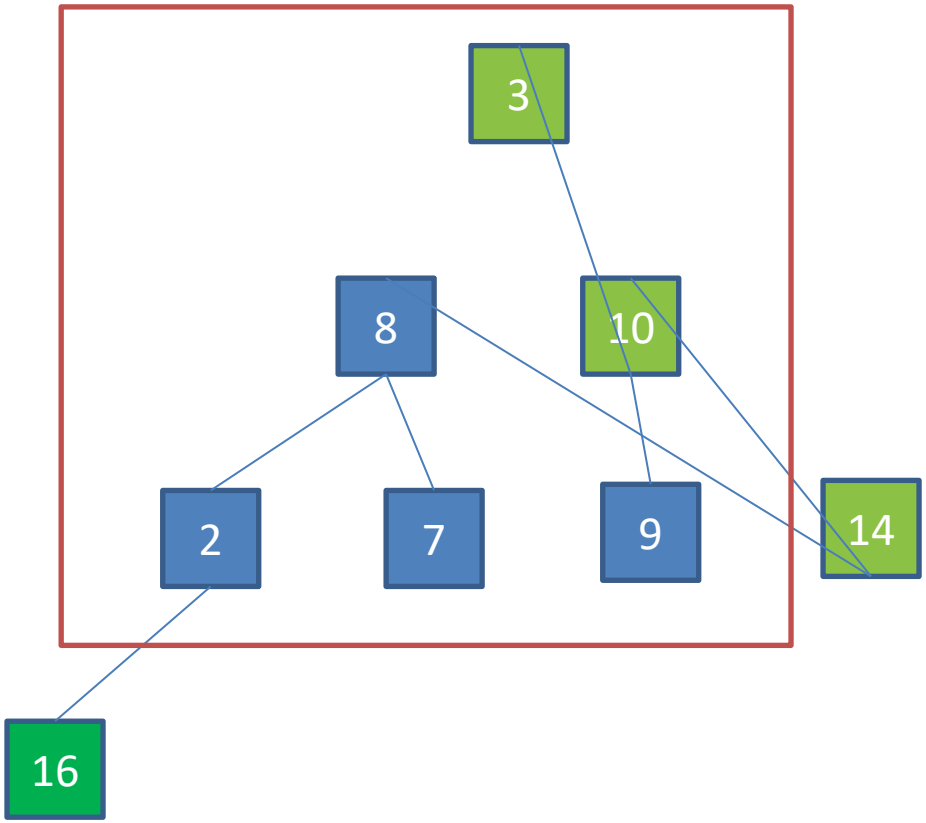
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



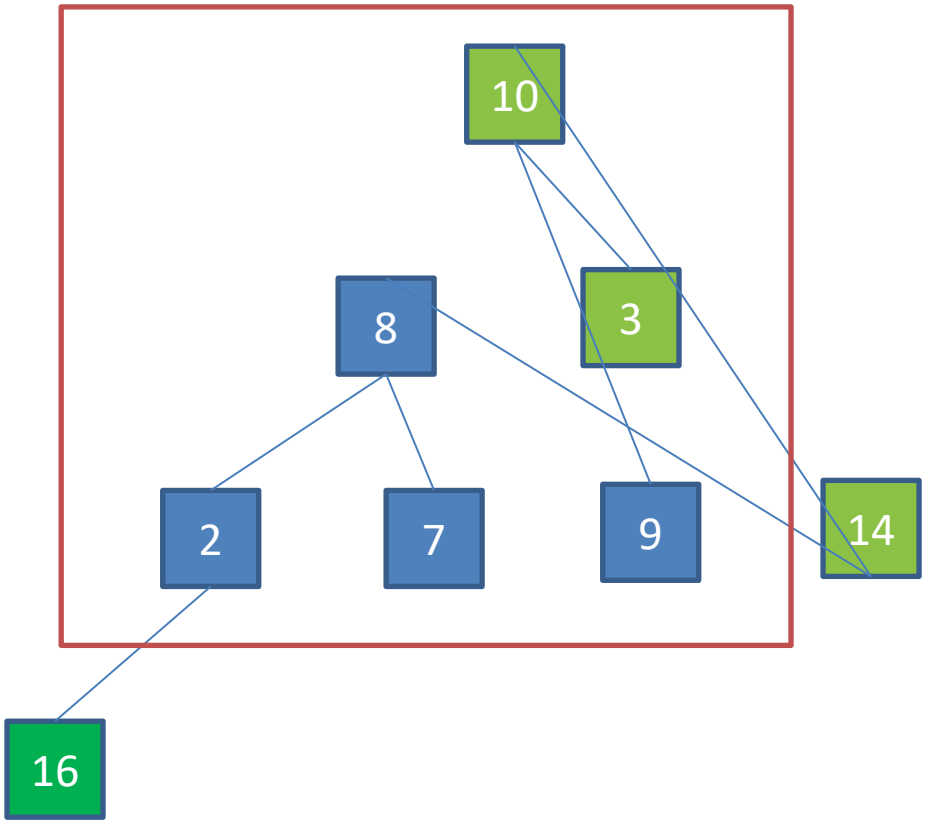
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



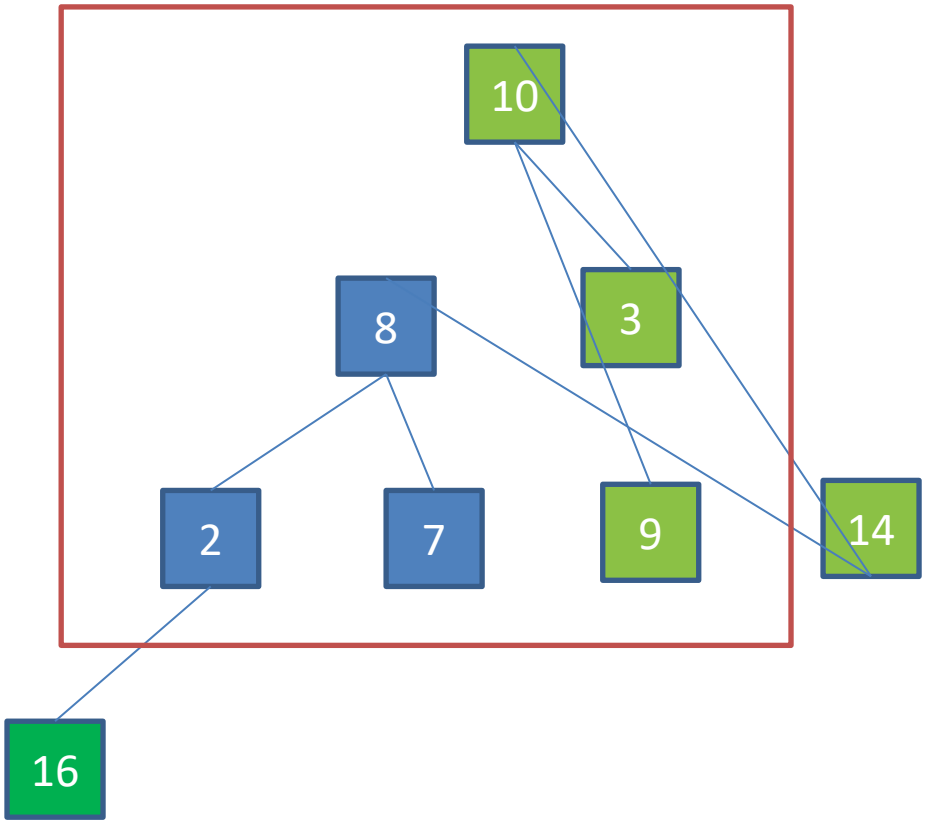
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



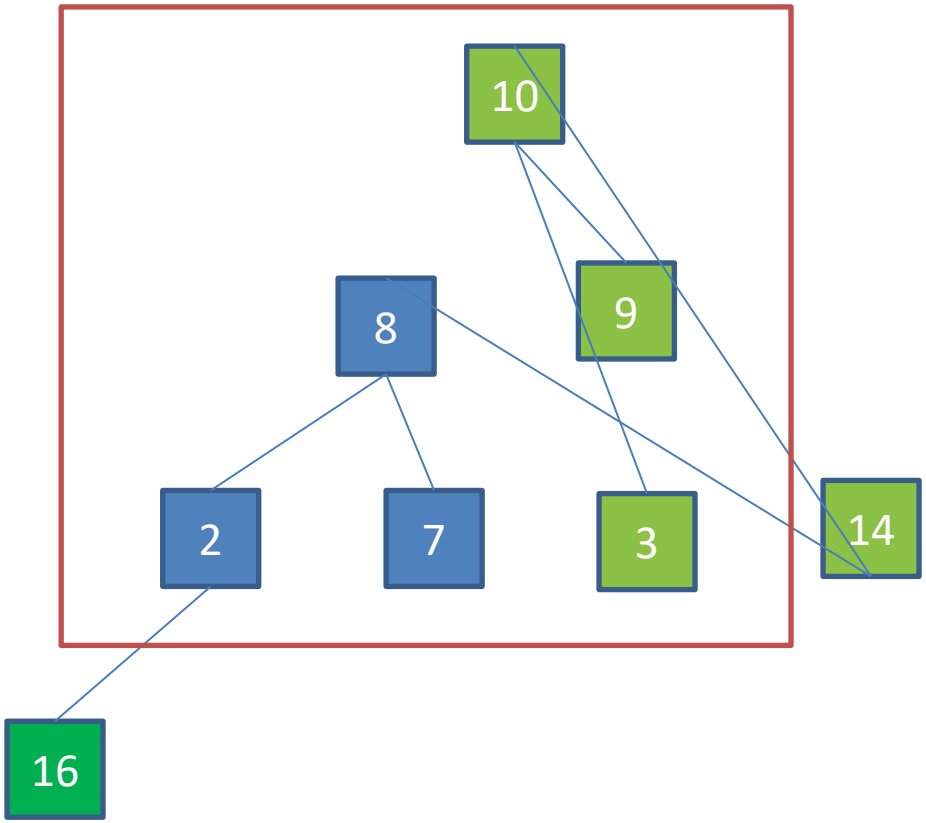
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



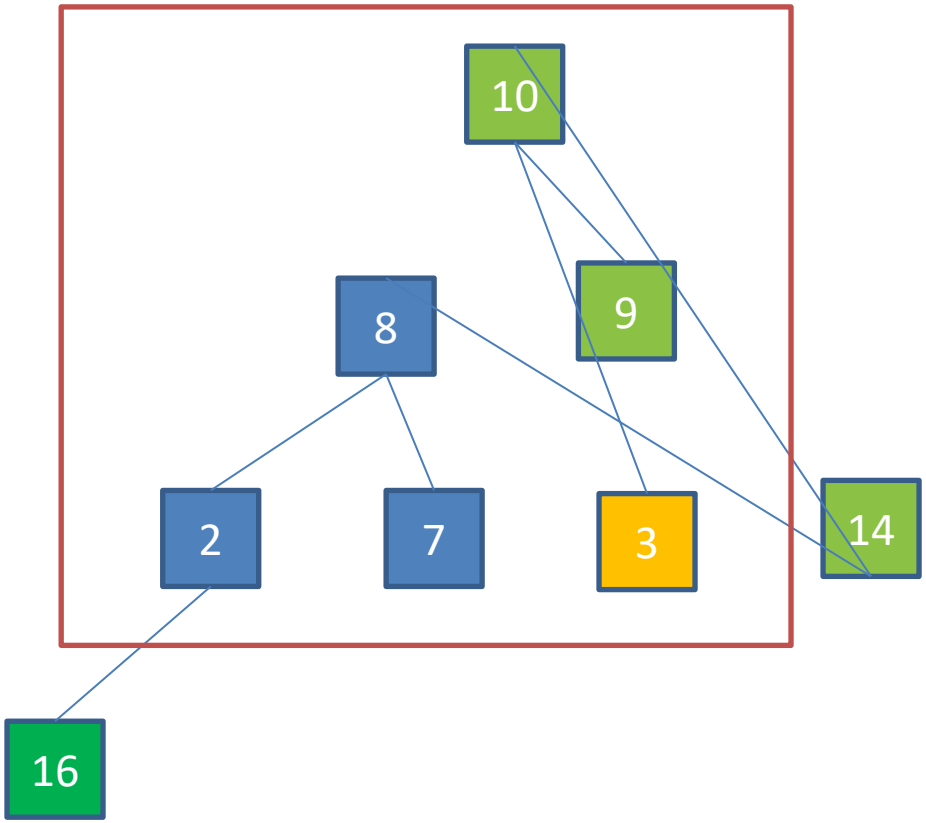
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



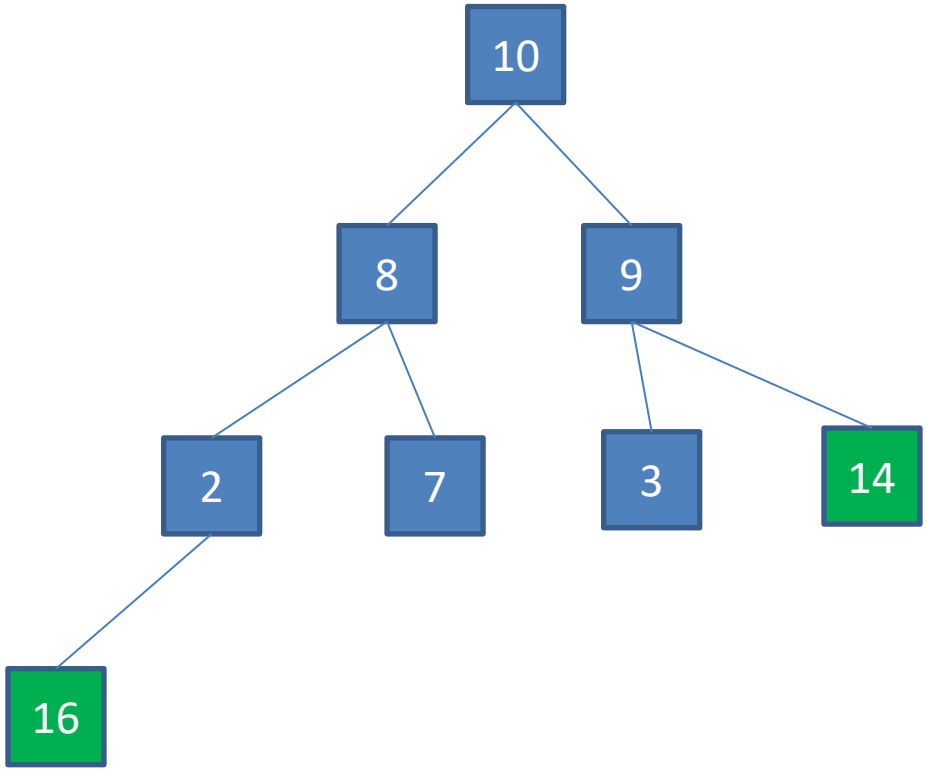
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



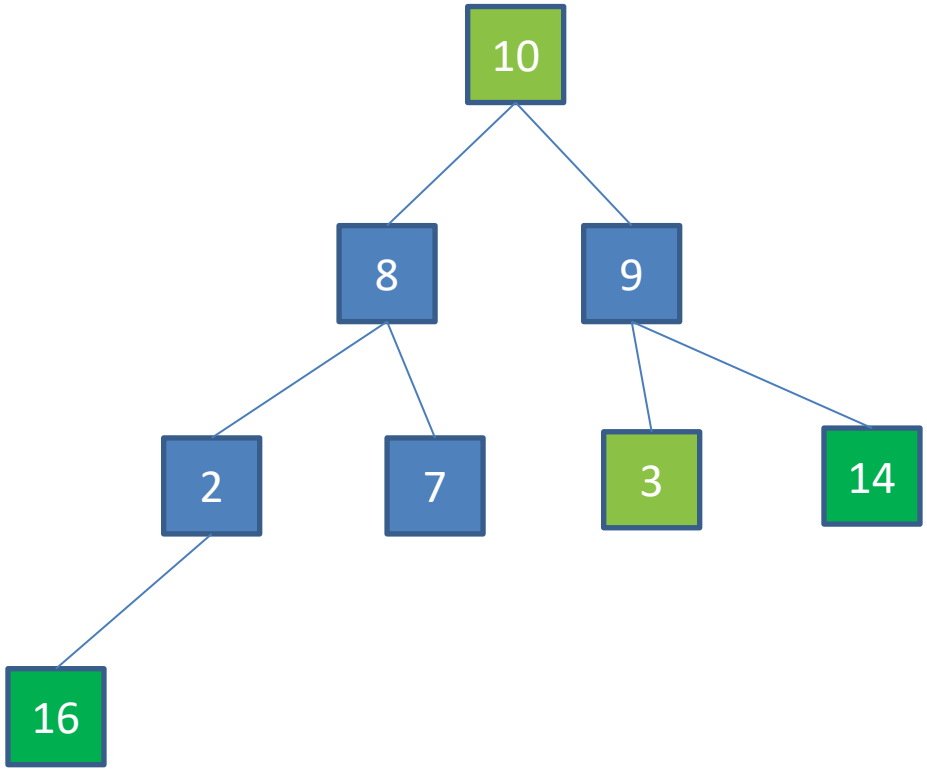
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



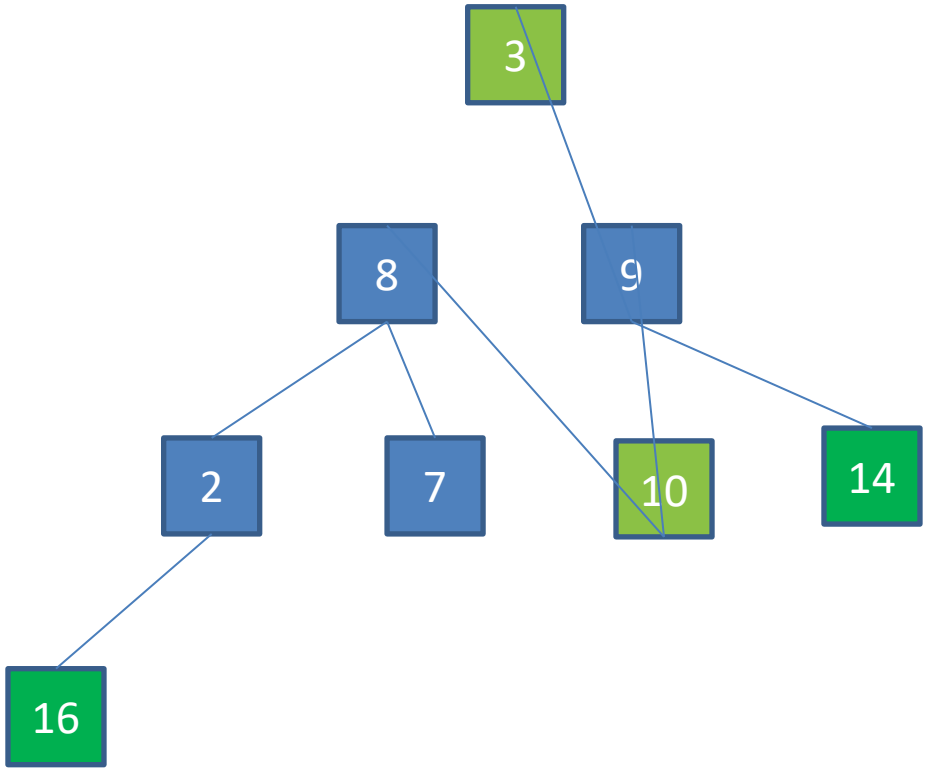
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



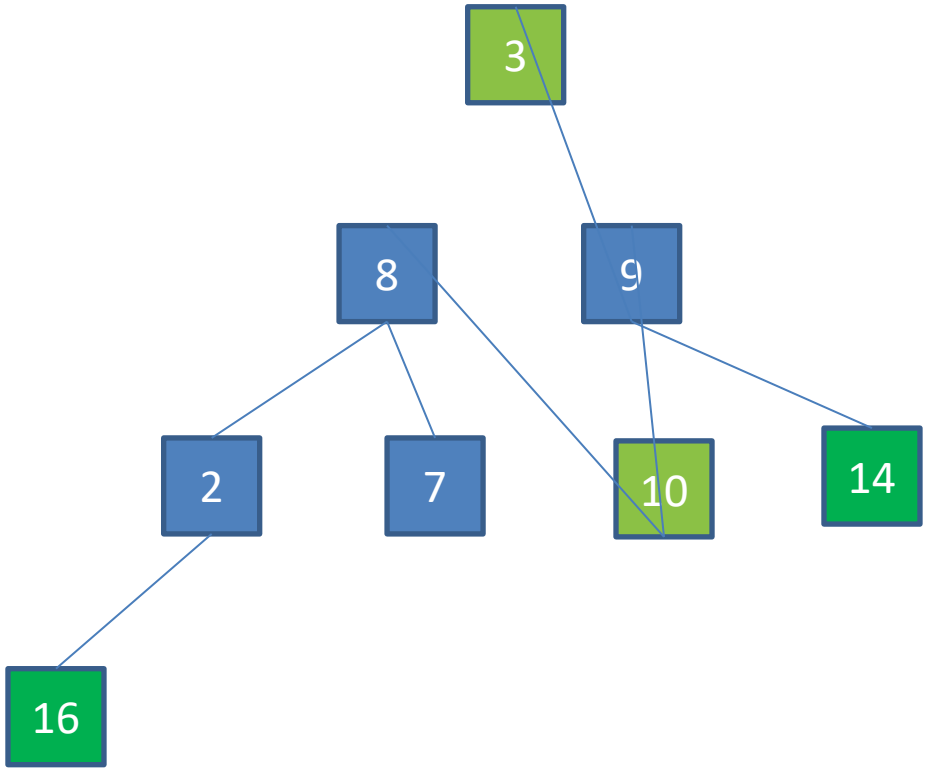
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



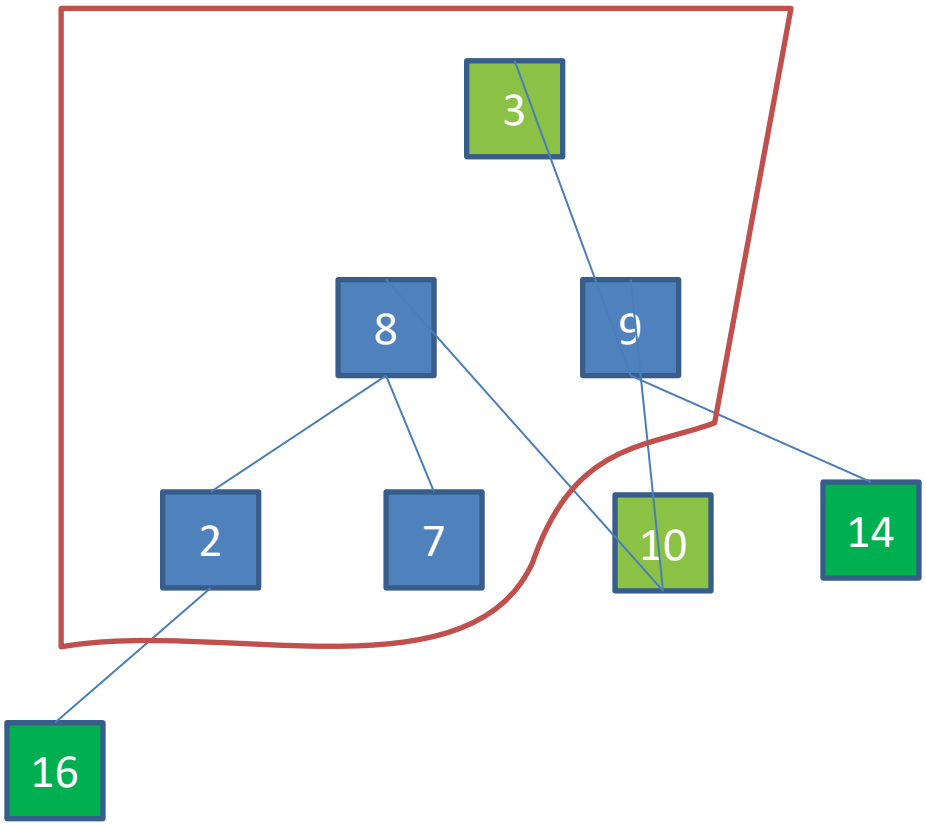
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



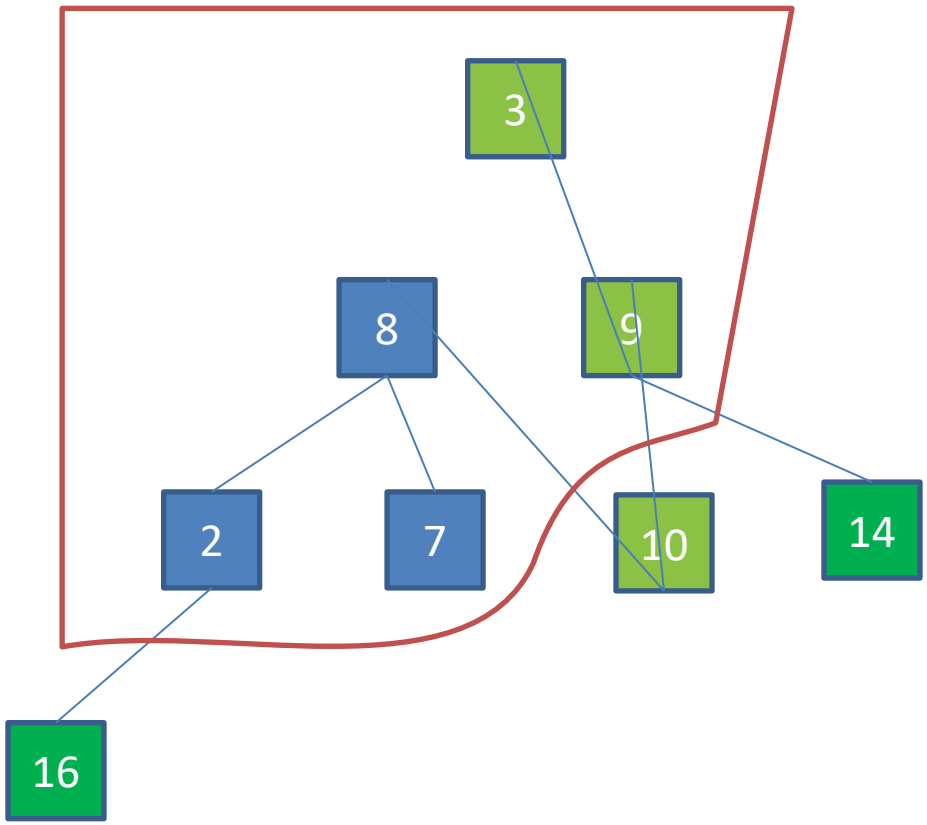
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



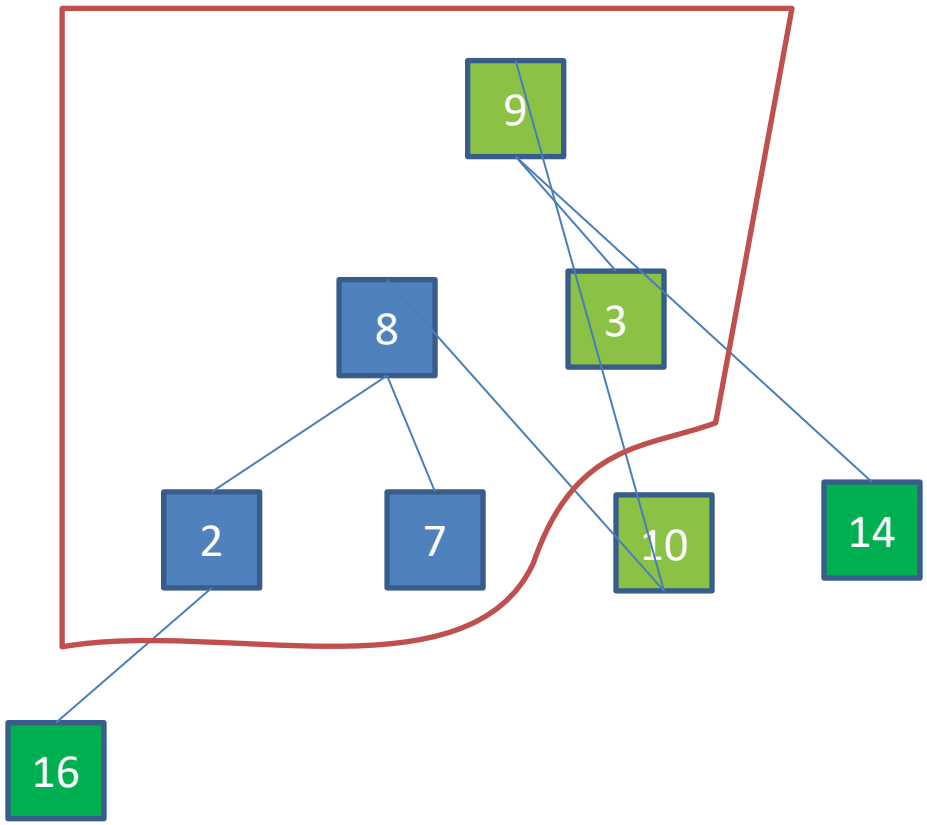
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



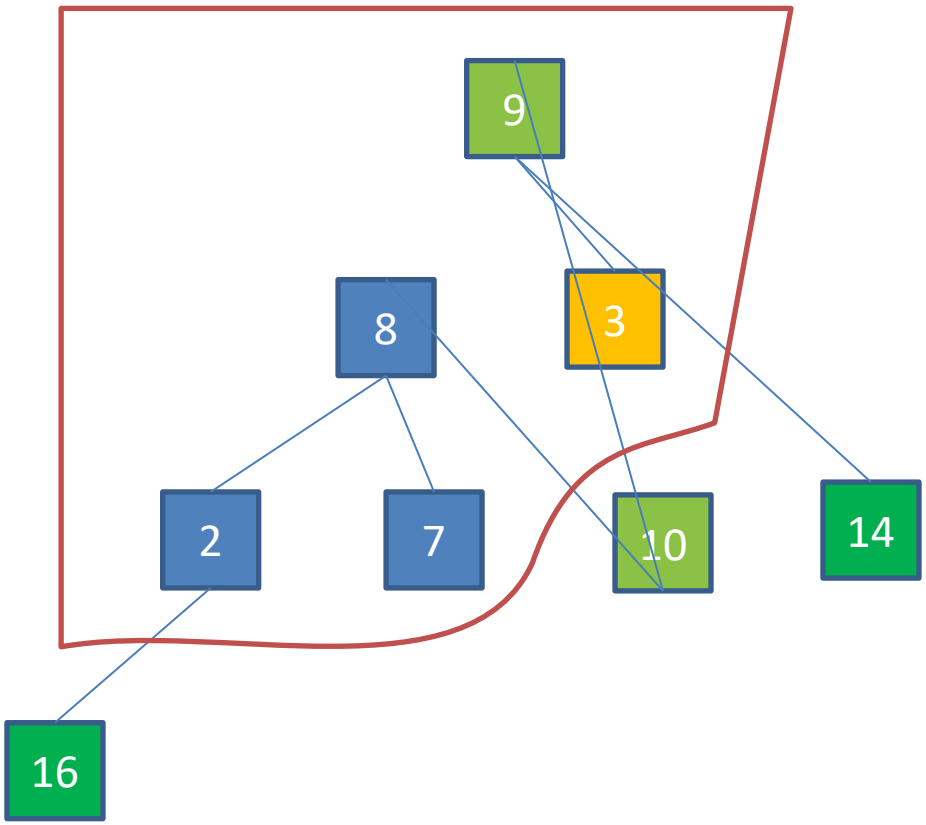
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



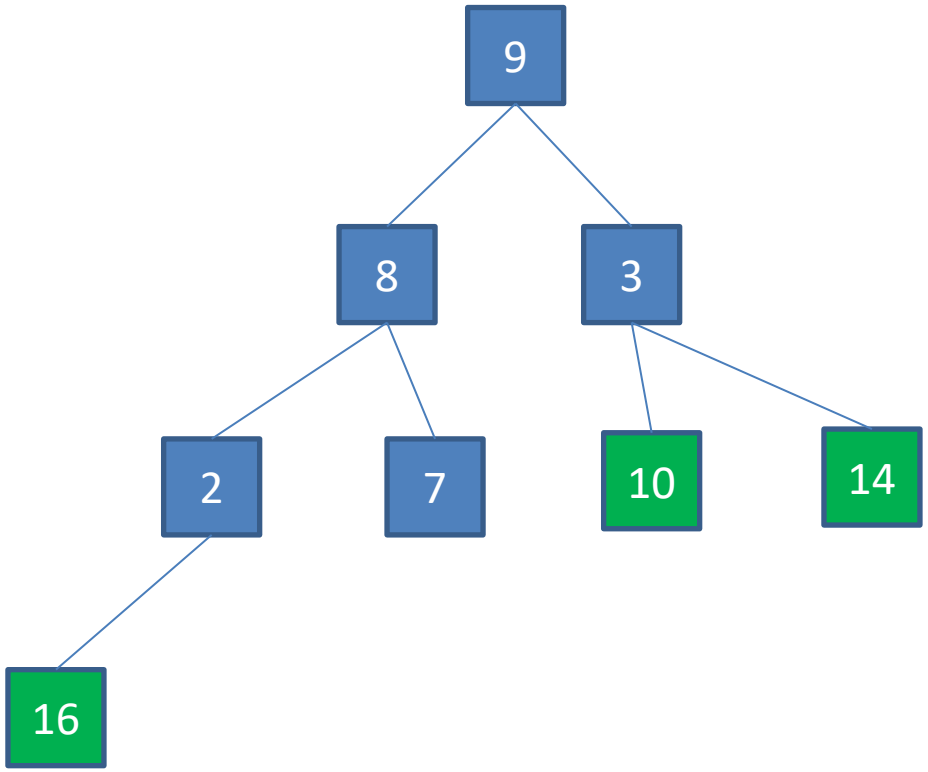
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



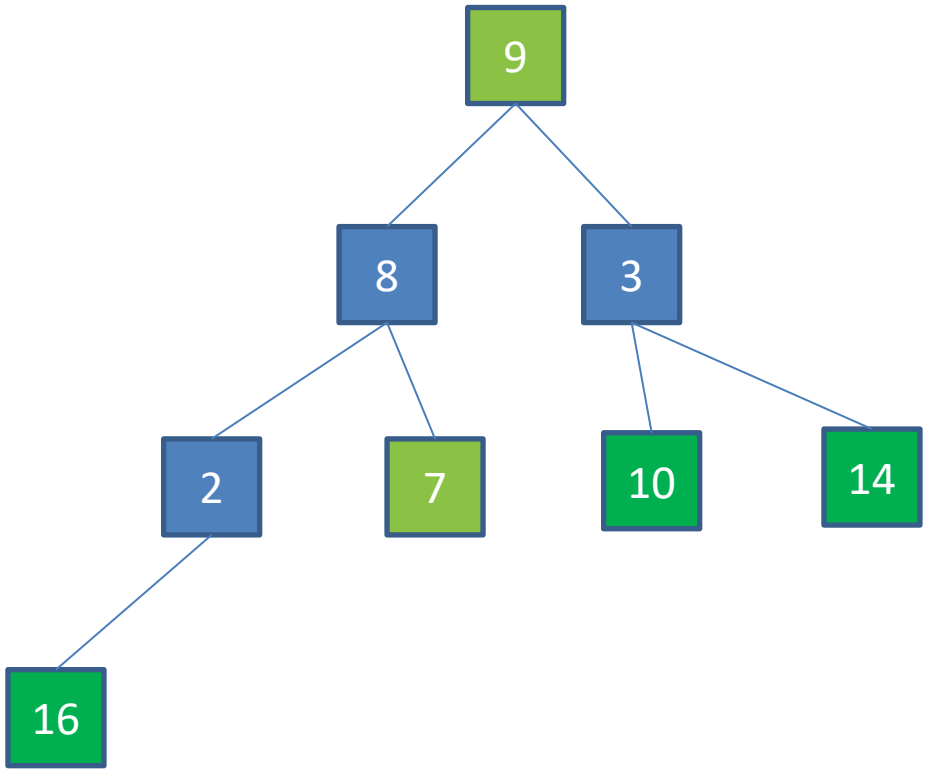
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



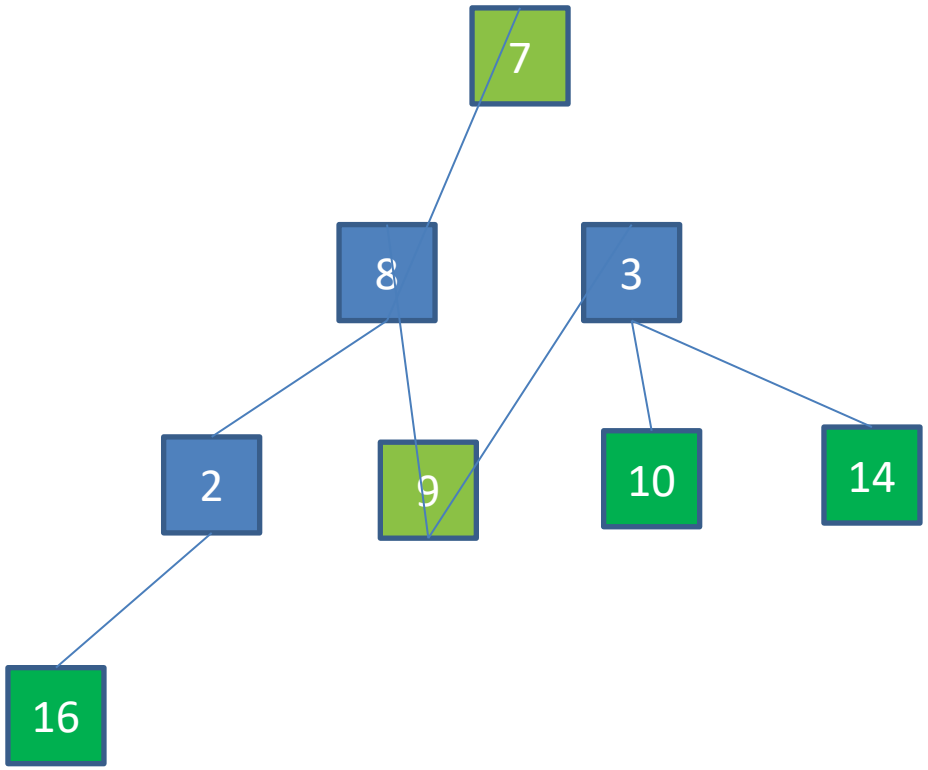
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



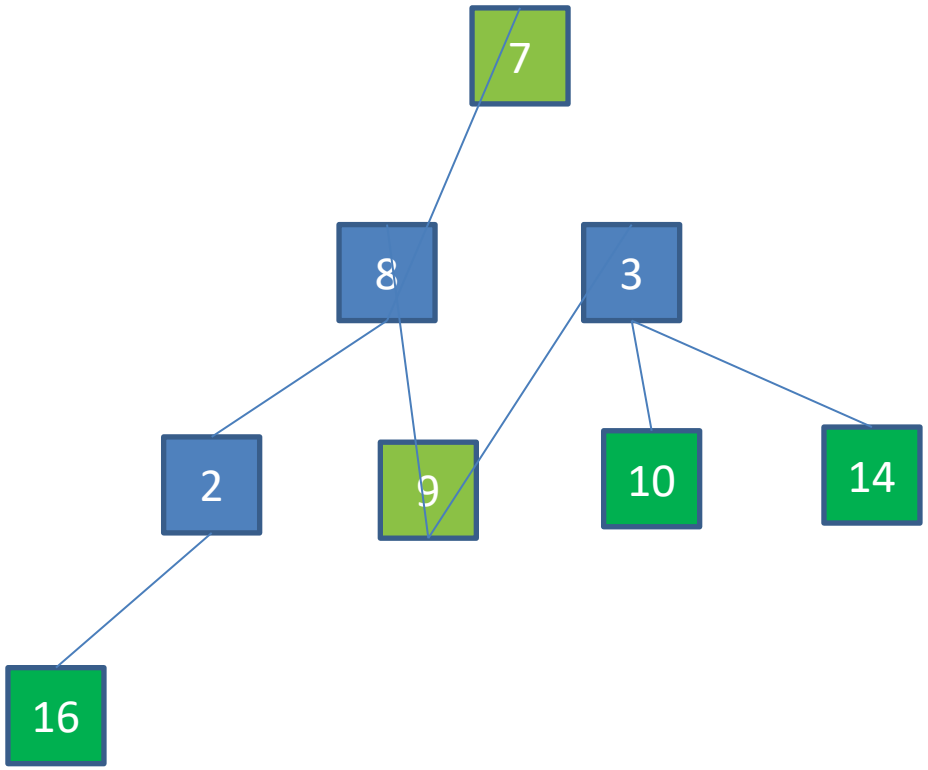
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



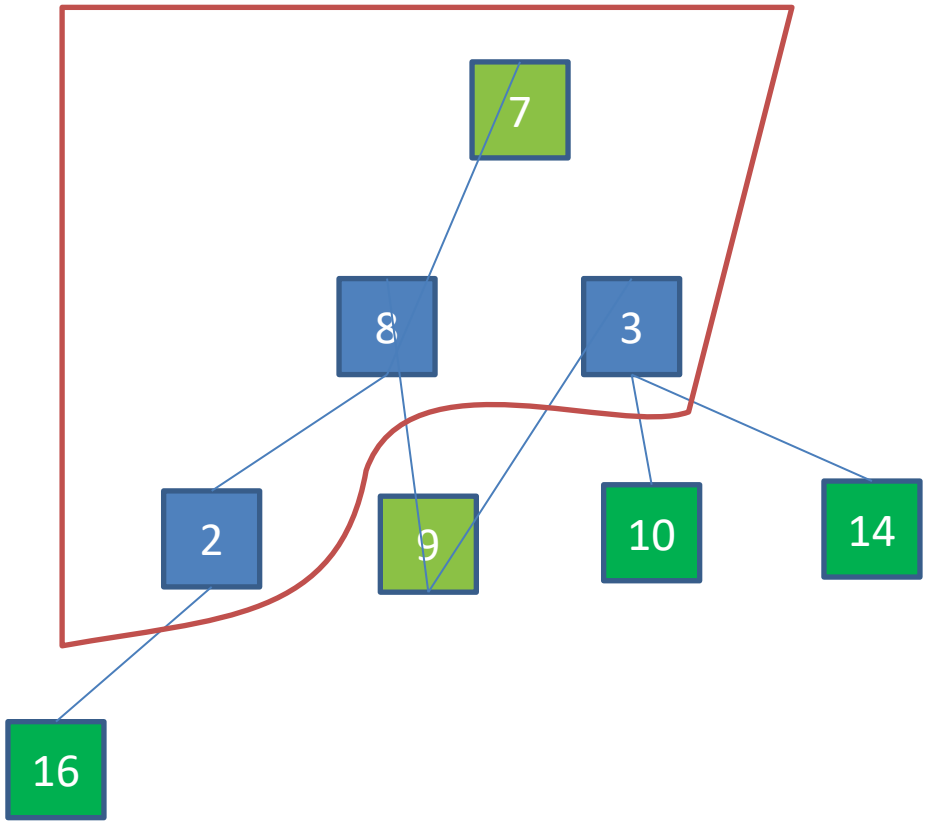
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



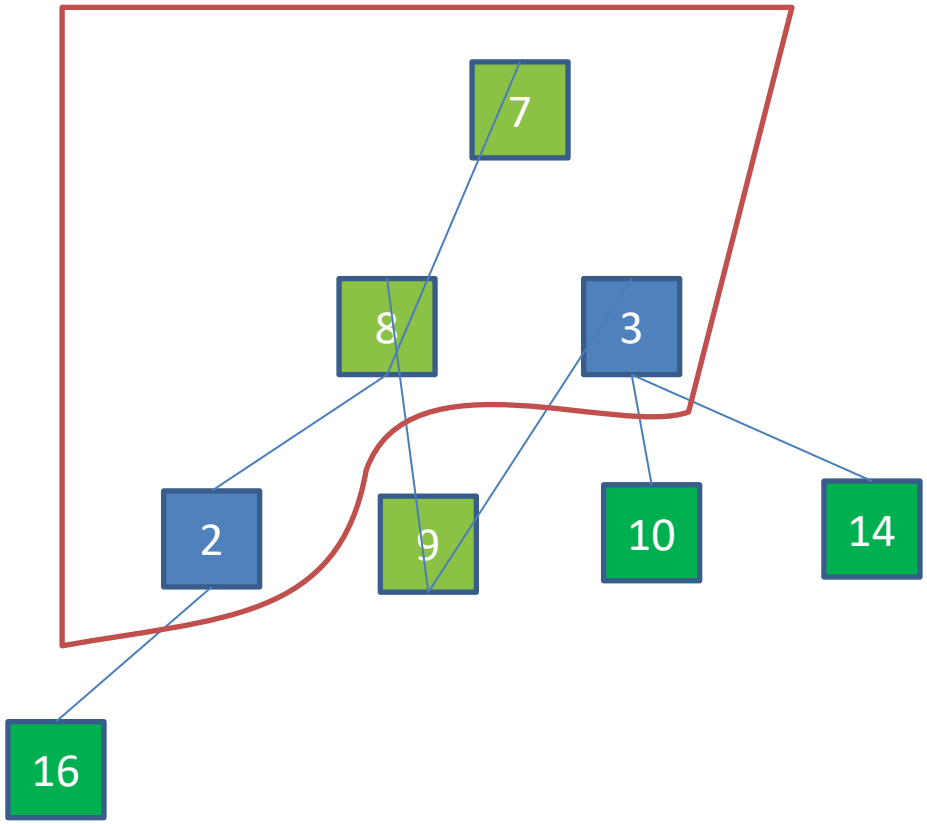
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



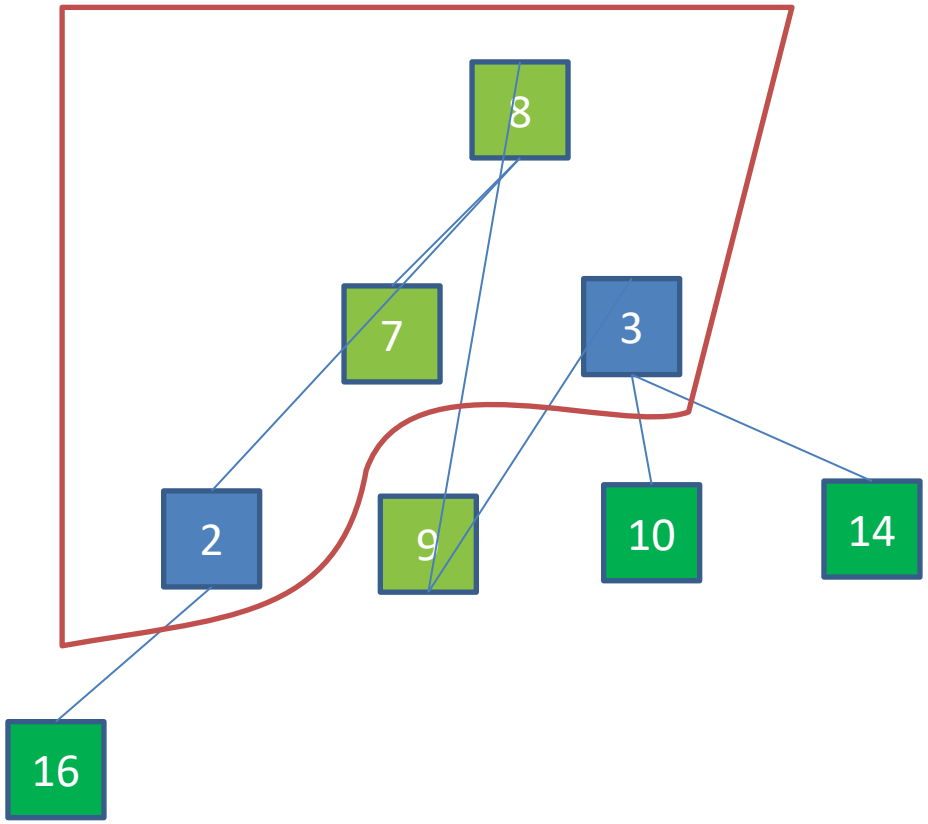
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



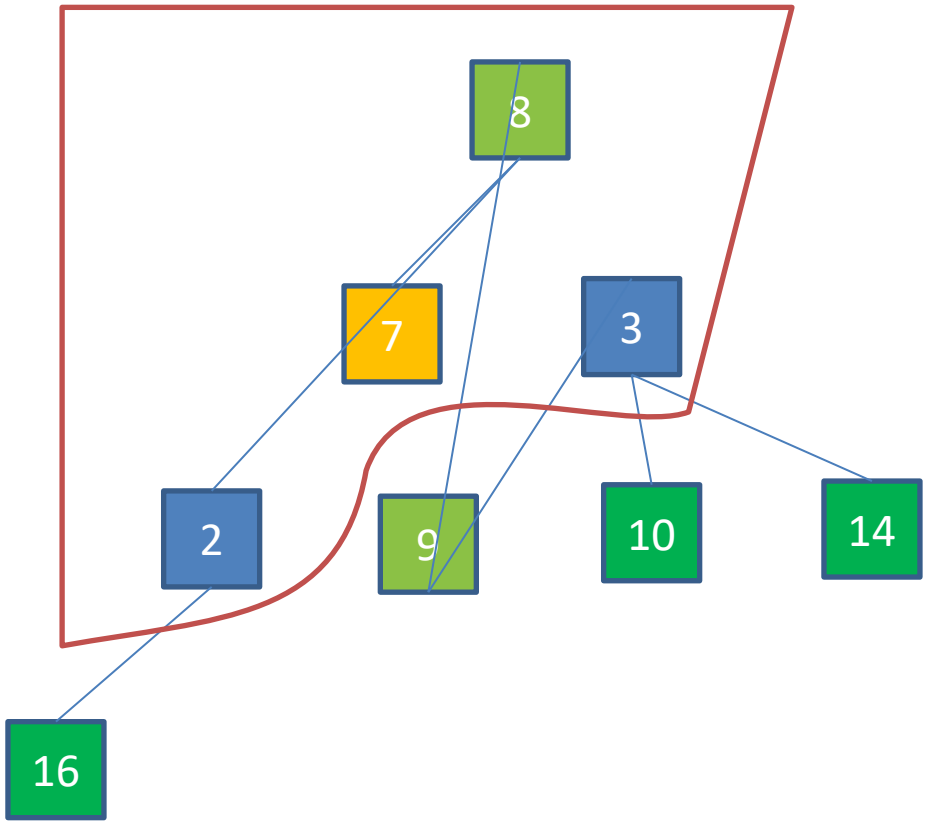
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



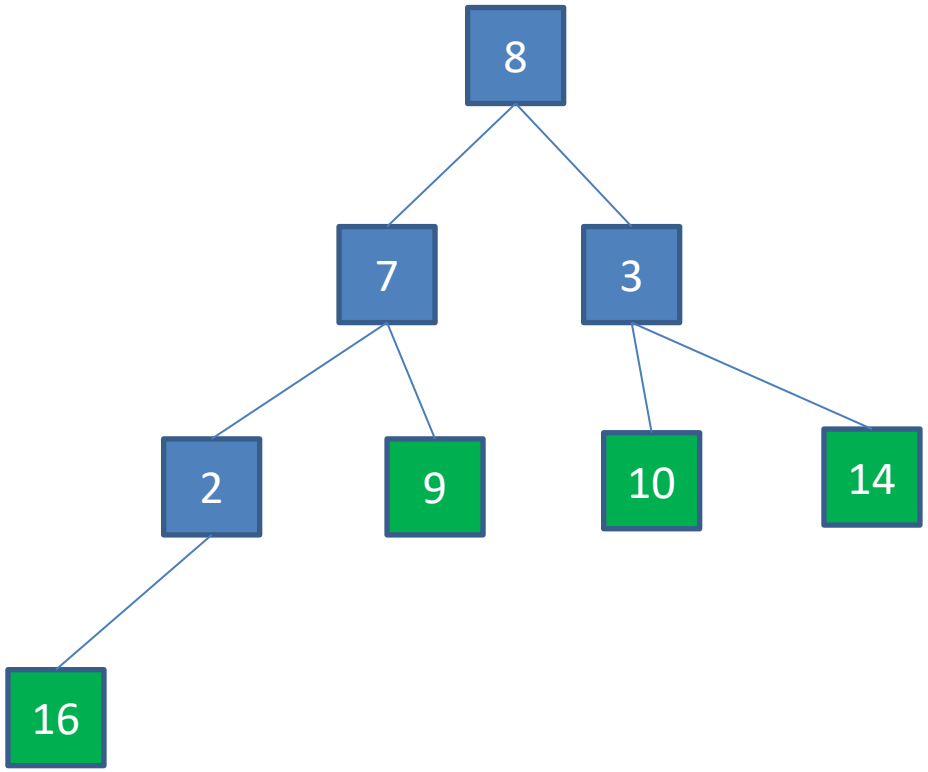
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



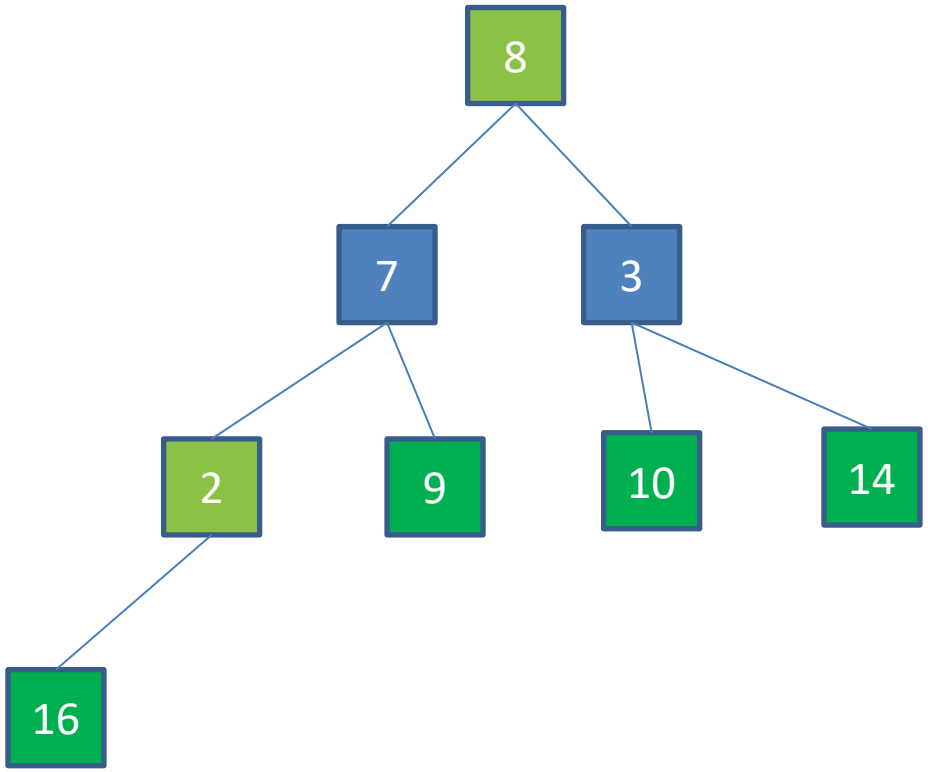
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



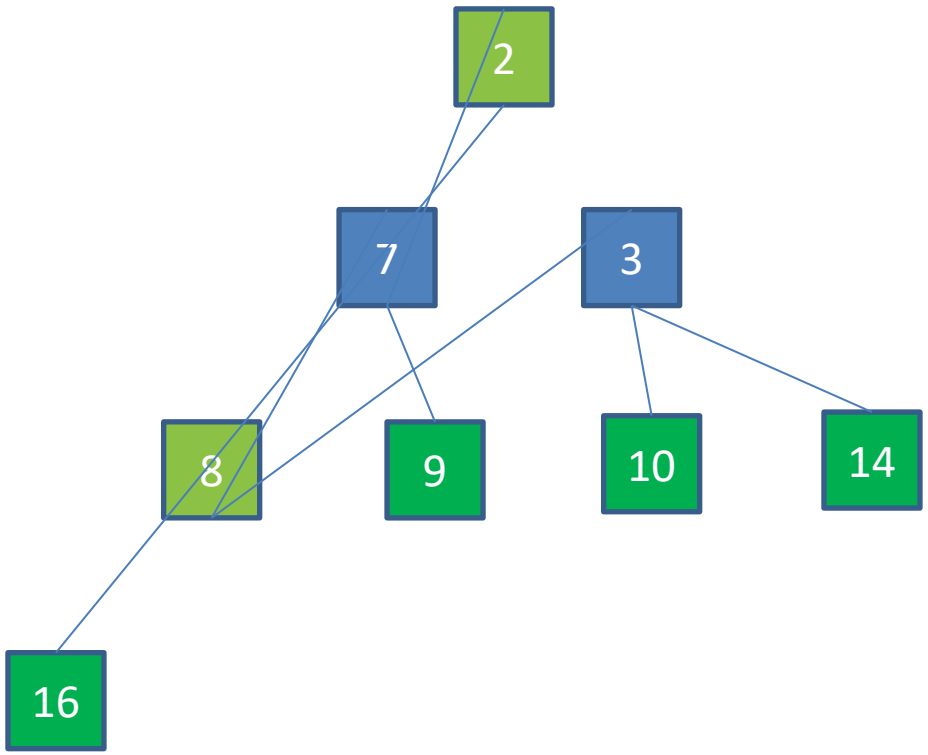
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



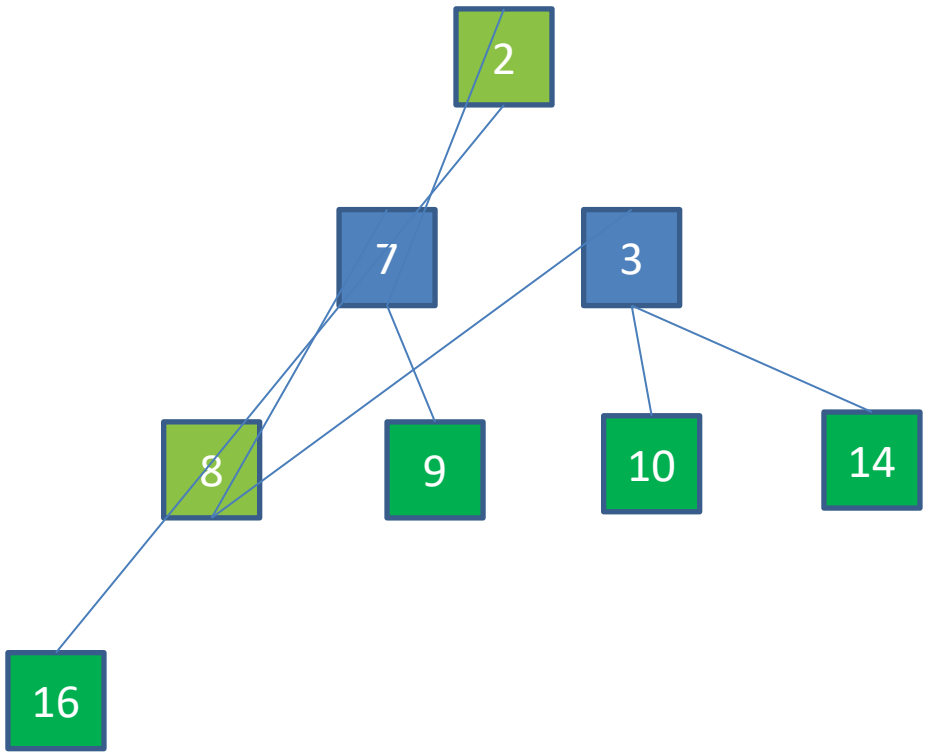
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



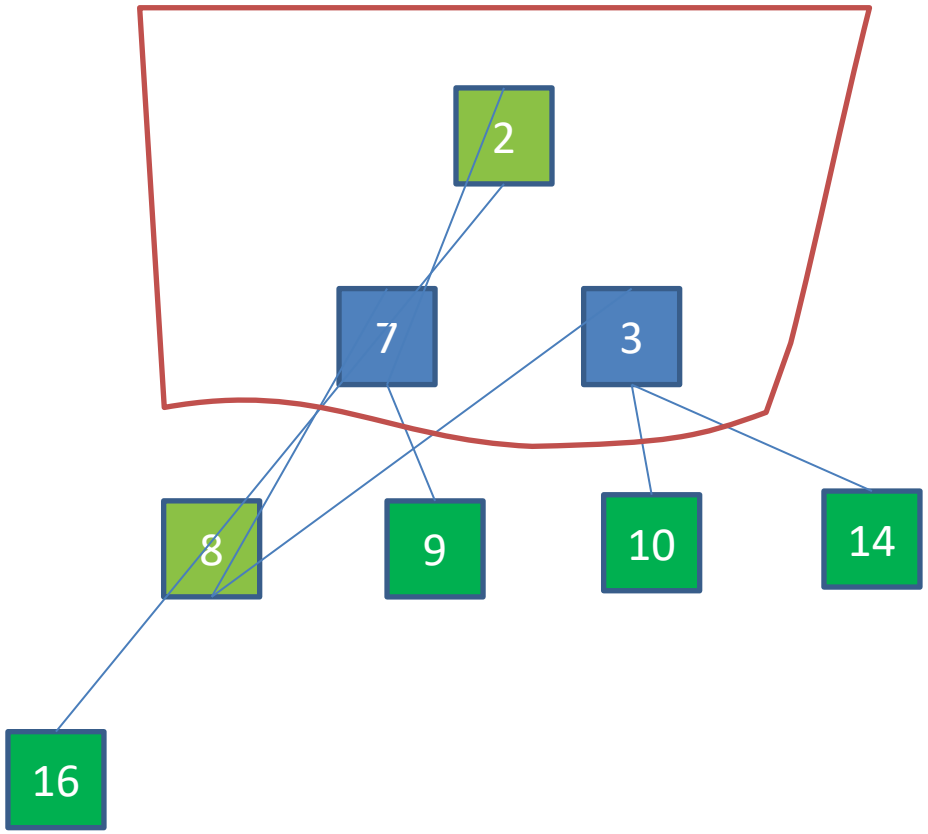
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



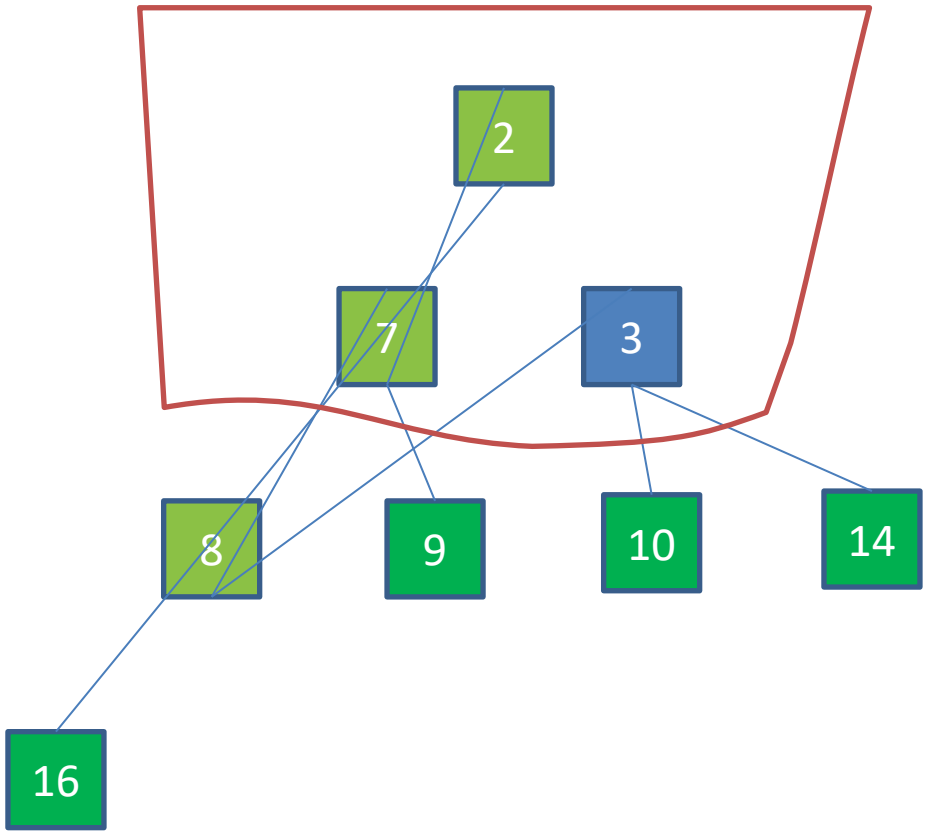
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



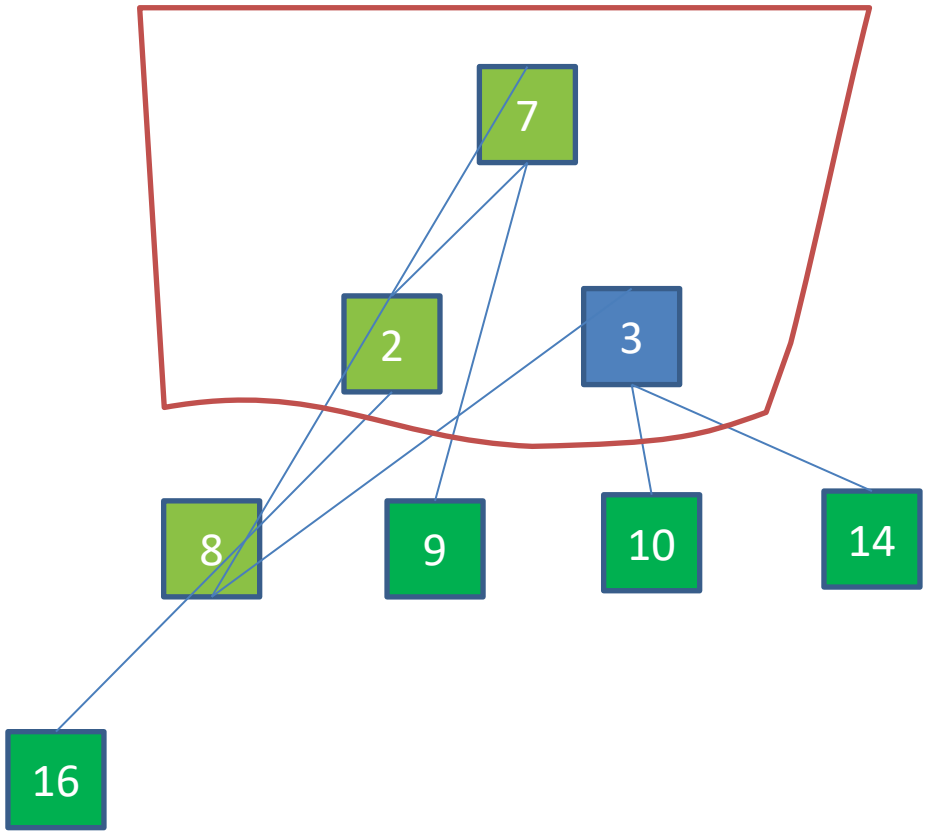
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



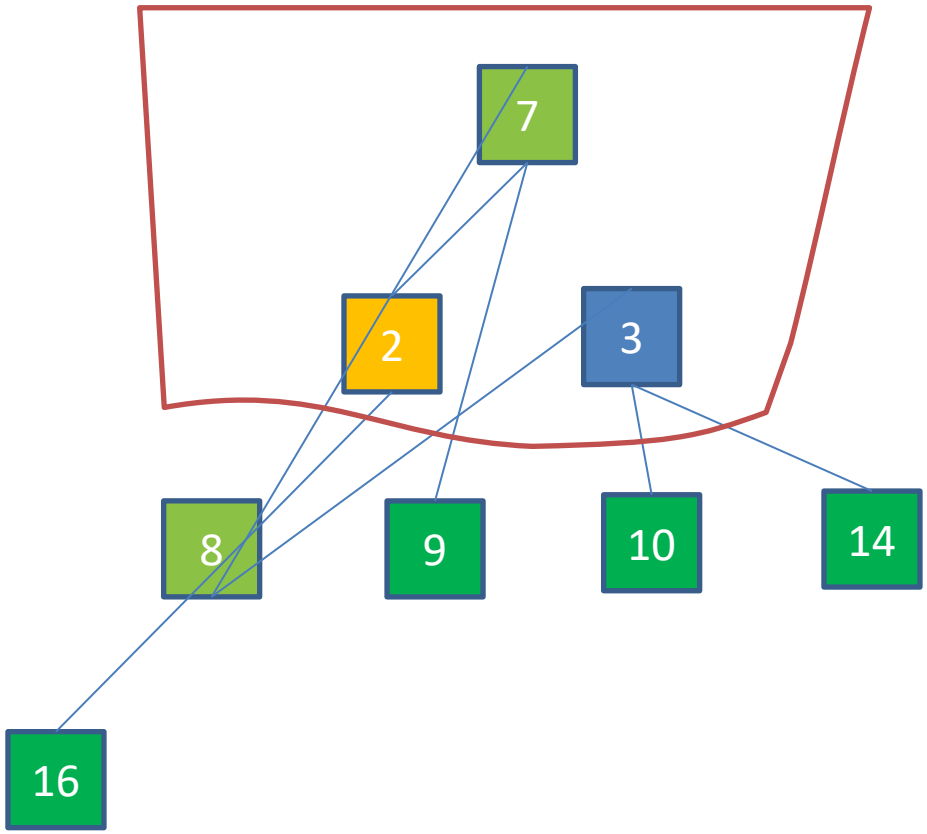
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



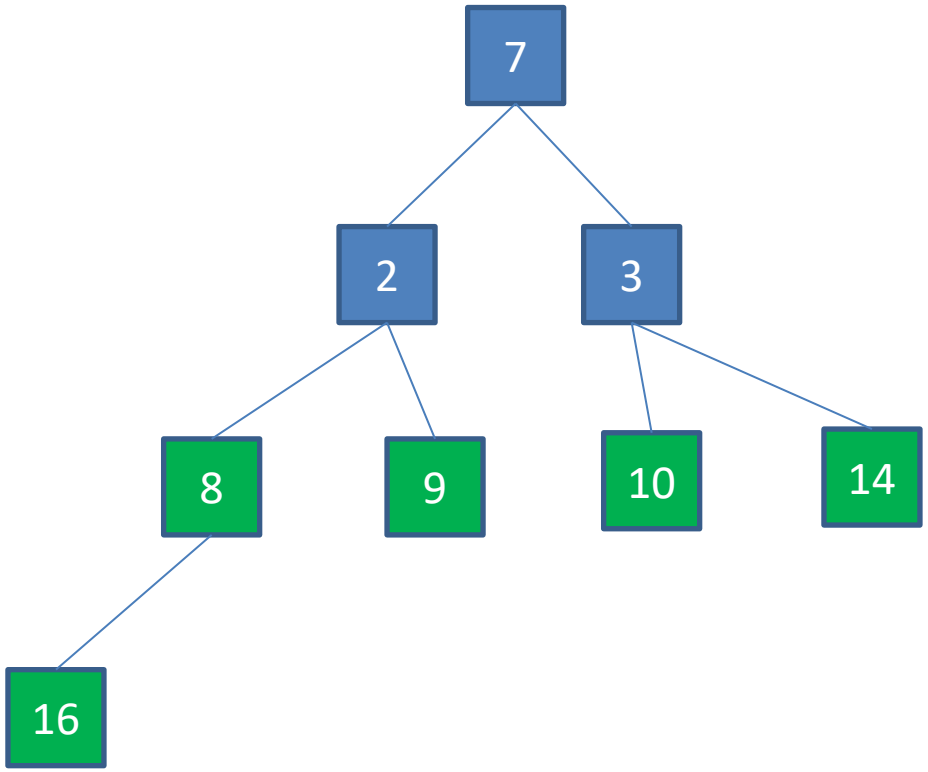
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



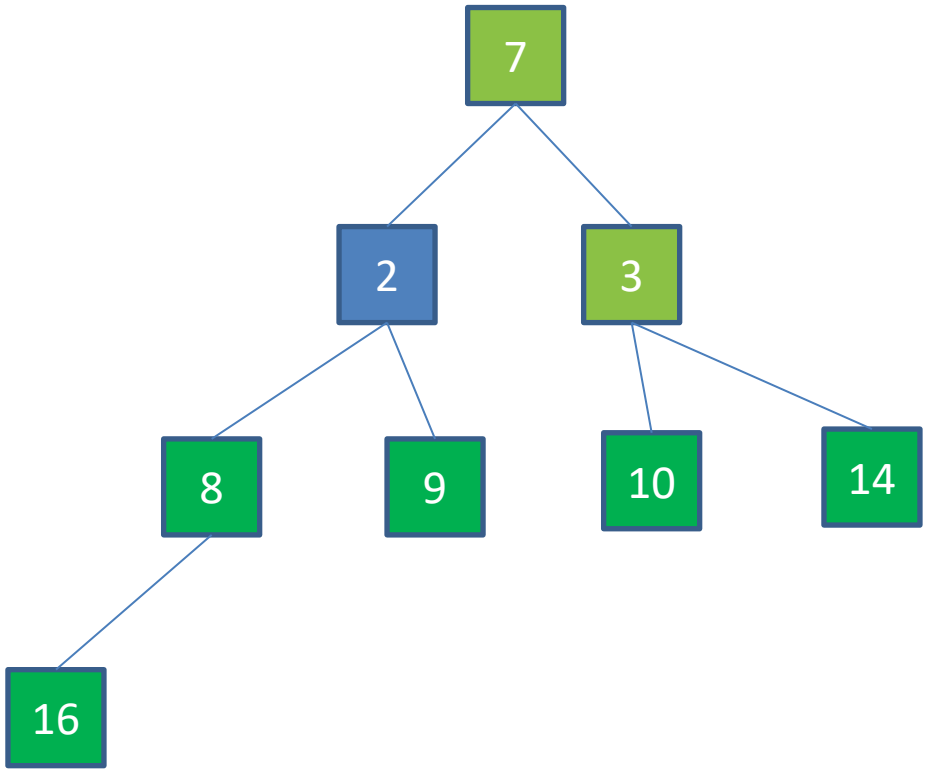
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



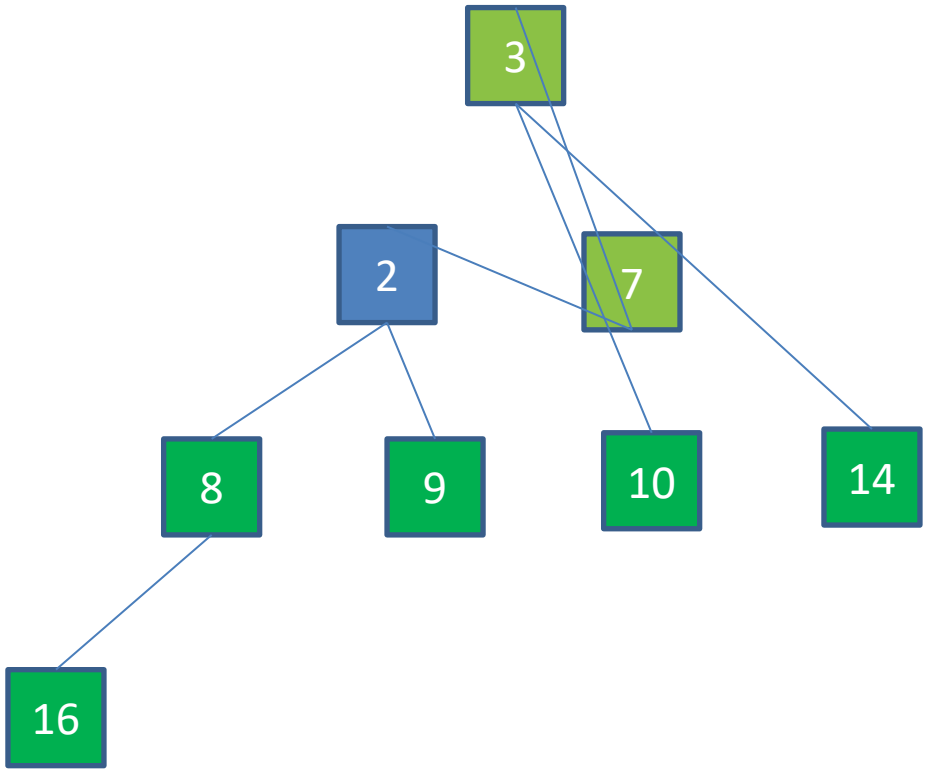
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



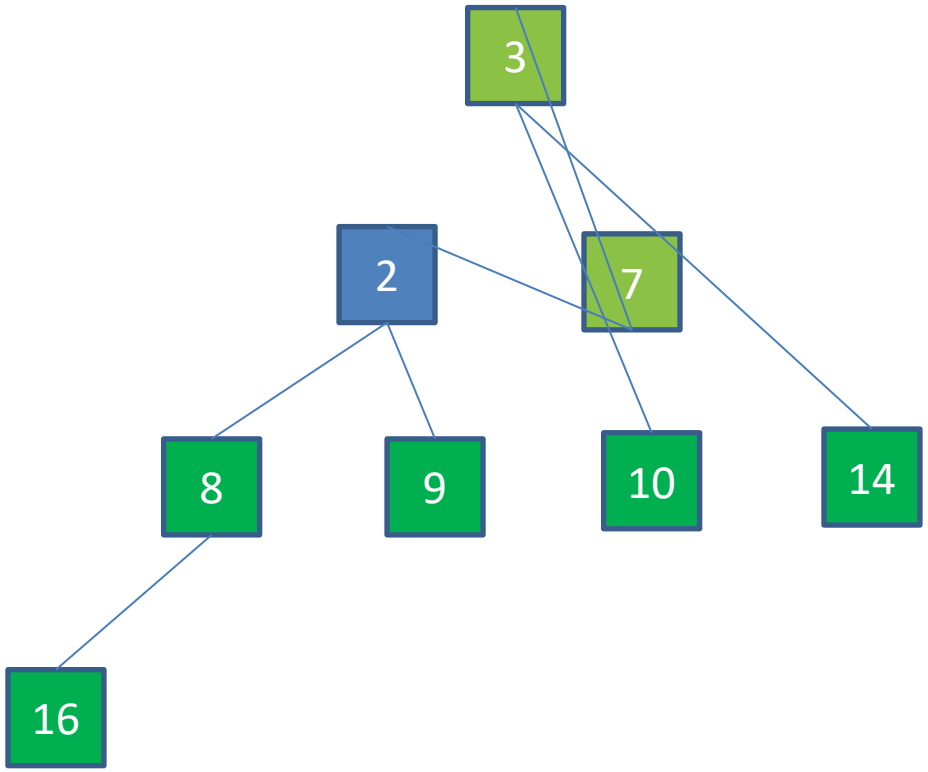
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



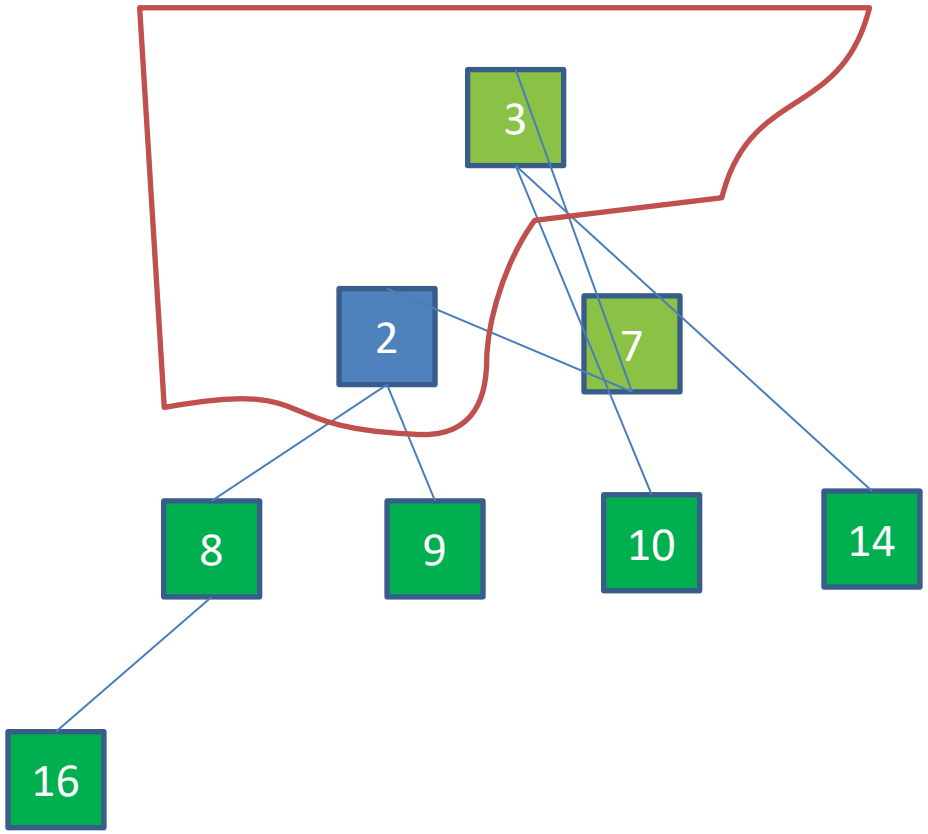
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



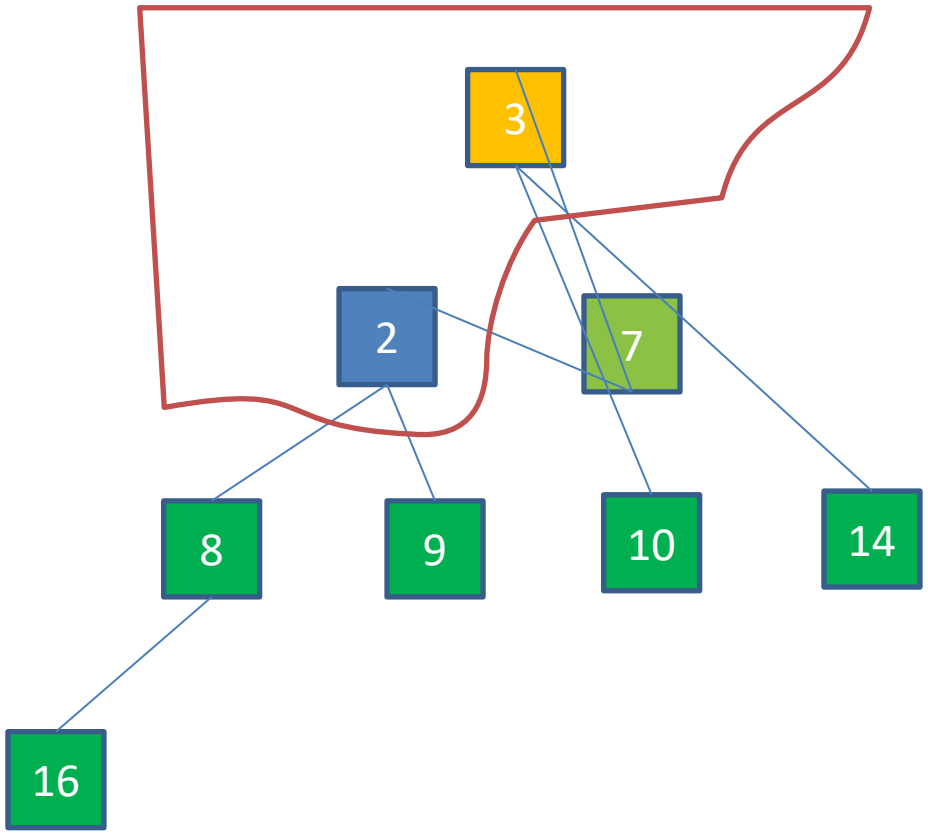
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



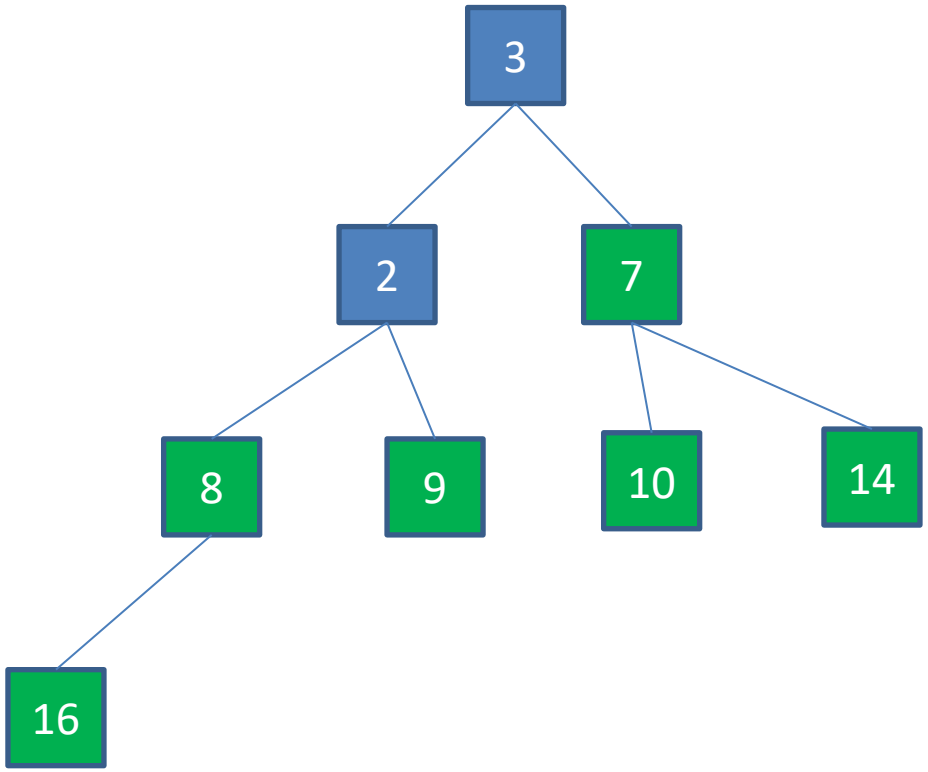
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



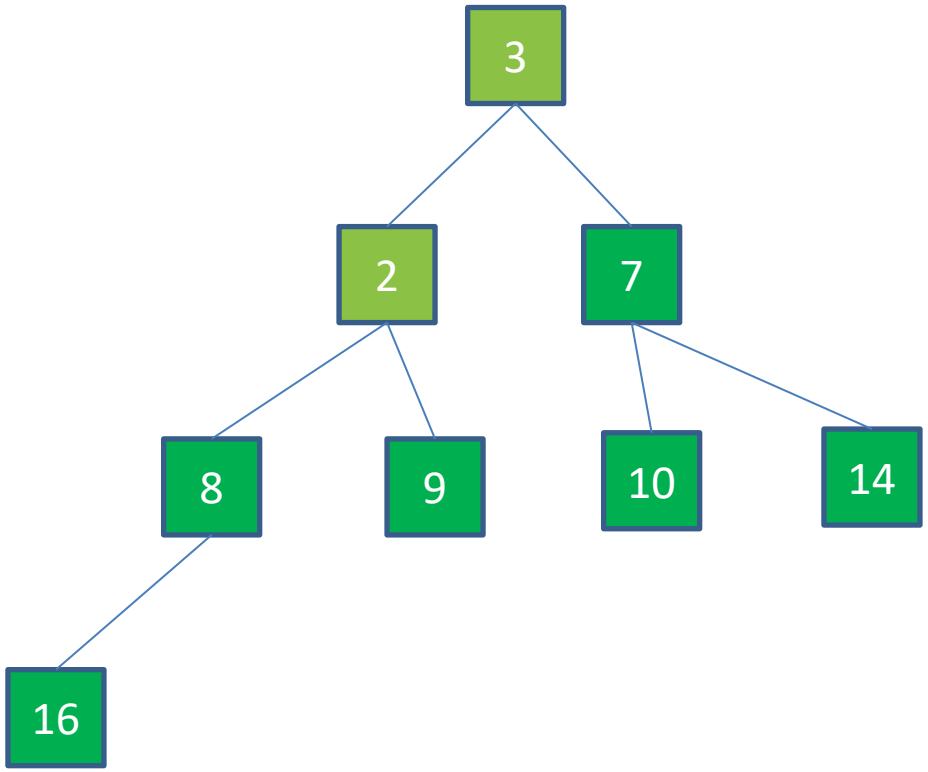
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



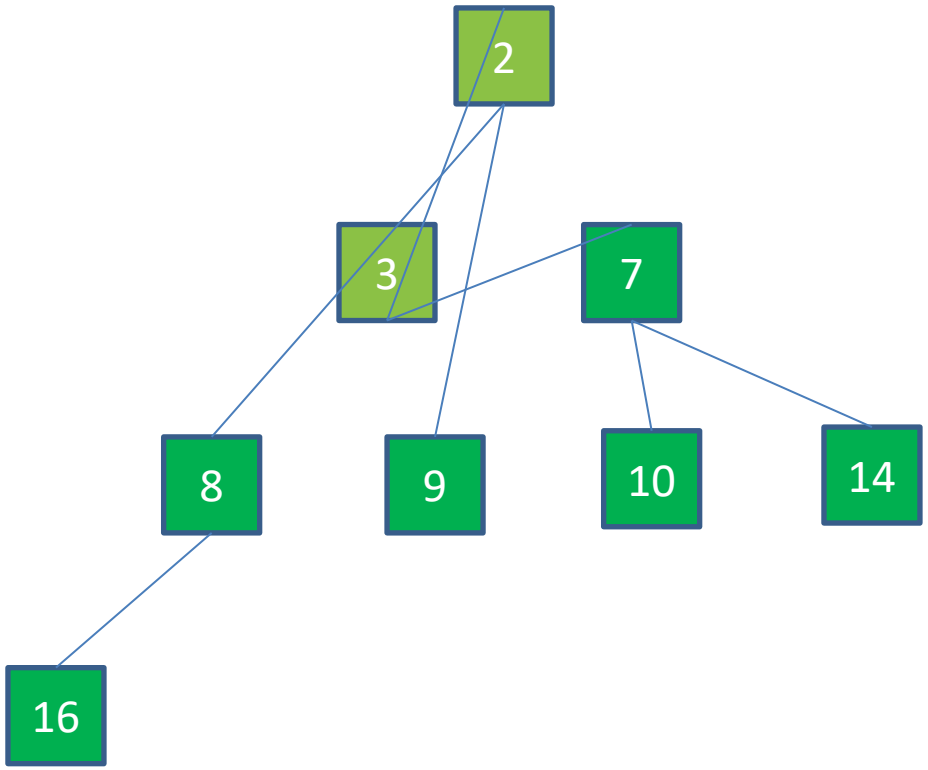
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



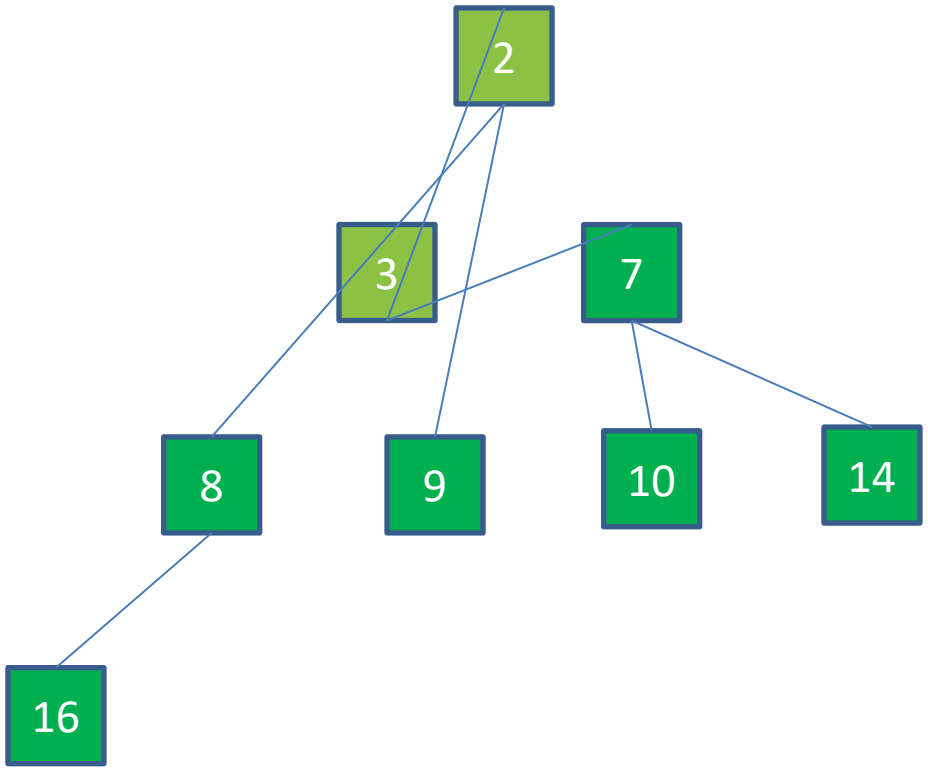
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



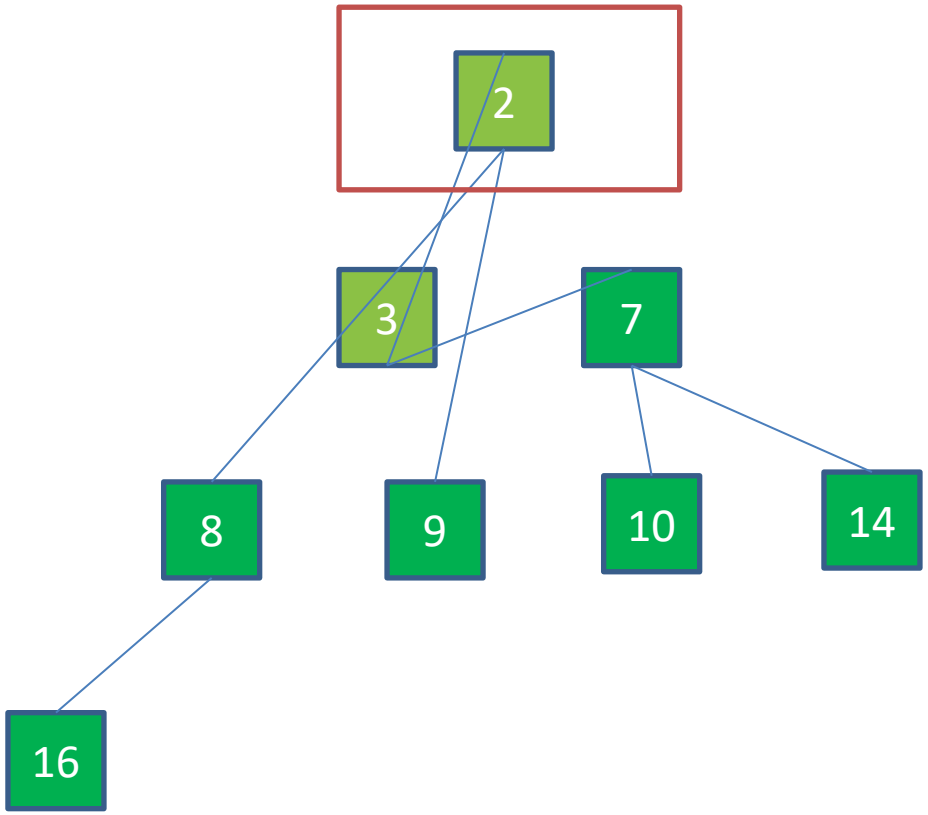
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



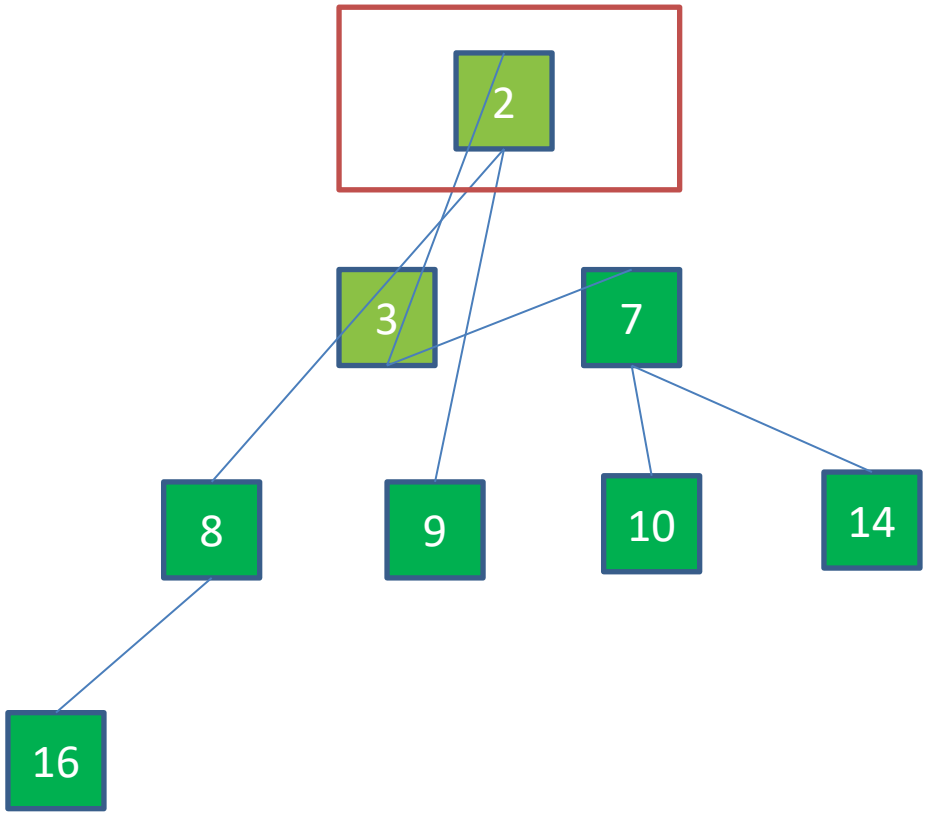
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



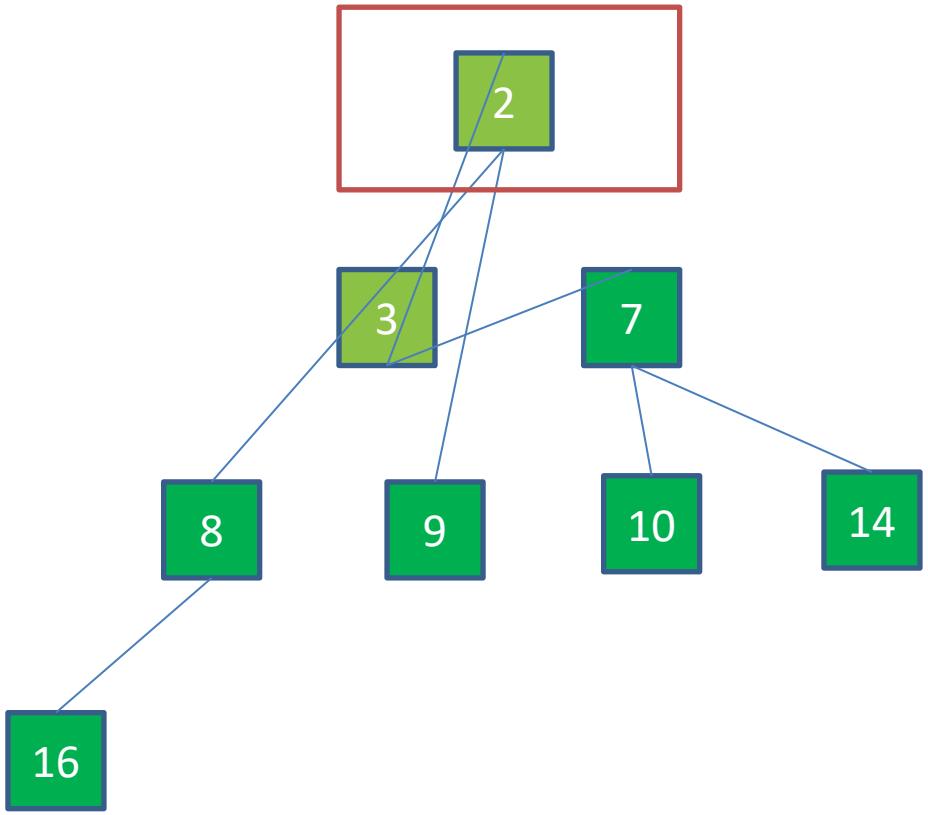
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



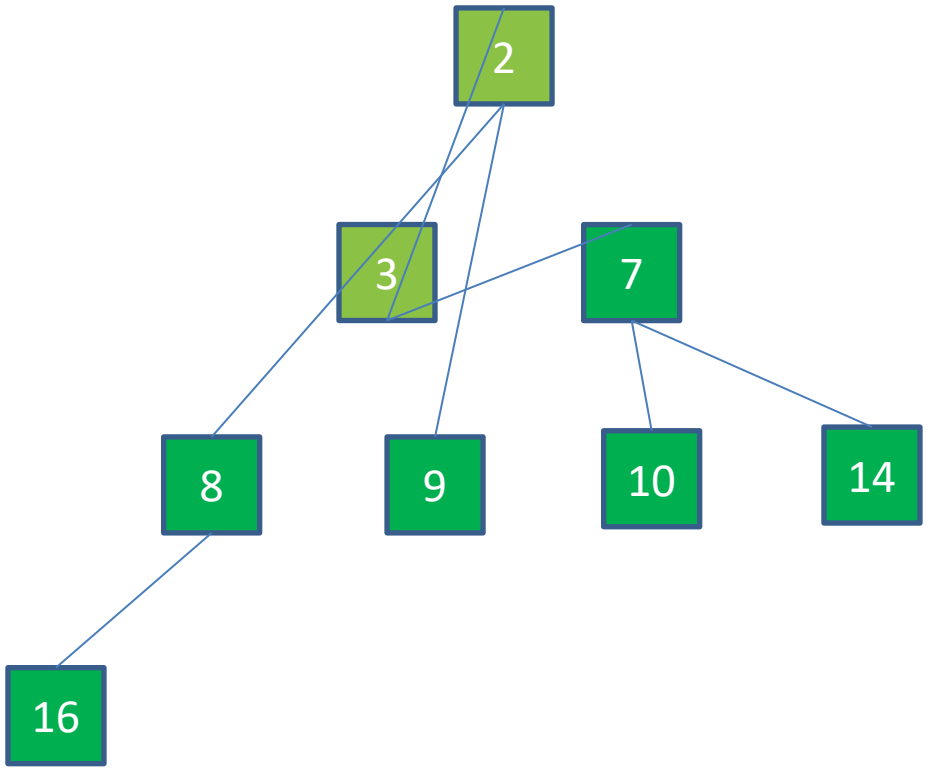
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



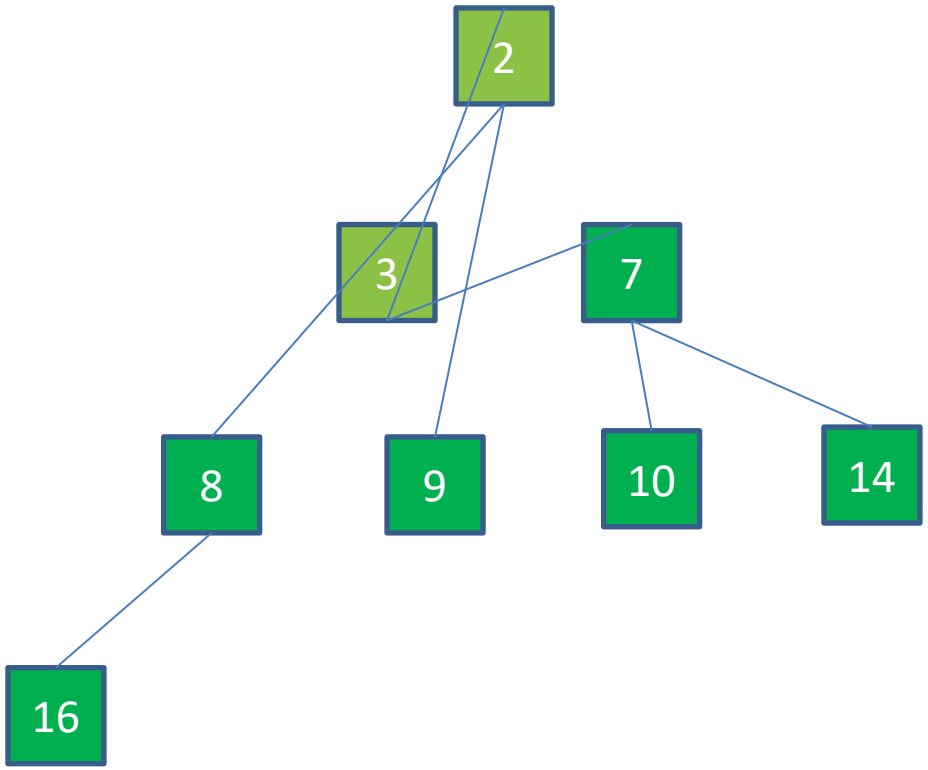
HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Heapsort Example



HEAP-SORT (A):

1. **BUILD-MAX-HEAP**(A)
2. Last node index i = A's last node index
3. From last element to the second in A {
   exchange (i, root);
   i--;
   MAX-HEAPIFY(A, root, i);
   }

MAX-HEAPIFY (A, i, t)

1. if(right(i)>t and left(i)>t) return;
2. Choose largest (node i, left(i), right(i))
3. if(the largest node is not i) {
   m = the index of the larger node
   Exchange i with the largest node
   MAX-HEAPIFY (A, m, t)
   }

Final result: a sorted array A