

Heap Sort

21

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology

21

i	0	1	2	3	4	5	6	7
a[i]	15	11	9	8	5	3	1	4

15

11

9

8

5

3

1

4

15

11

9

8

5

3

1

4

Easily find the max

Another way to find the root

root
index = 1

index = $2 \times 1 + 1 = 3$

index = $2 \times 3 + 1 = 7$

index = $2 \times 3 + 1 = 7$

4 5 3 2 0 1

2 3 4 5 6 7

AC2050: Data Structures & Algorithms
SEEE, Hanoi Univ. of Science and Technology

22

Algorithm

- ▶ **Build a binary max heap**
- ▶ Repeat until the array is sorted:
 - ▶ Swap the max (first) and the last element in the array
 - ▶ Reduce the data size by 1
 - ▶ Correct the heap because of the root node's new value
- ▶ Attn:
 - ▶ In a max heap, the root element has the max value
 - ▶ The first element in the array corresponds to the root in the tree

▶ 23

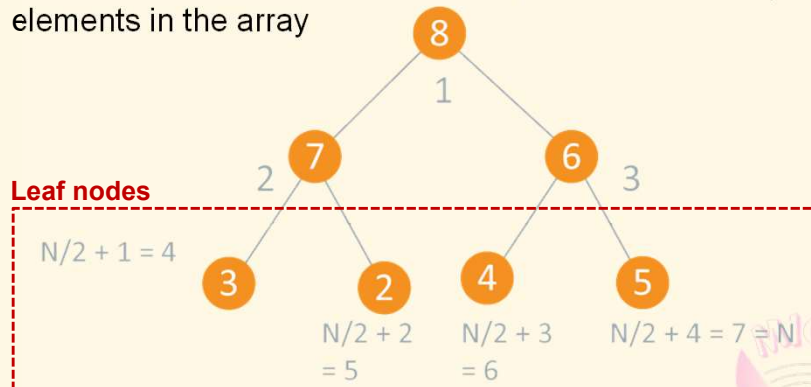
AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



23

Building Max Heap

- ▶ For each node, make sure that its children are smaller
- ▶ Check from lower nodes up to the root
- ▶ No need to check leaf nodes → Only check the first $n/2$ elements in the array

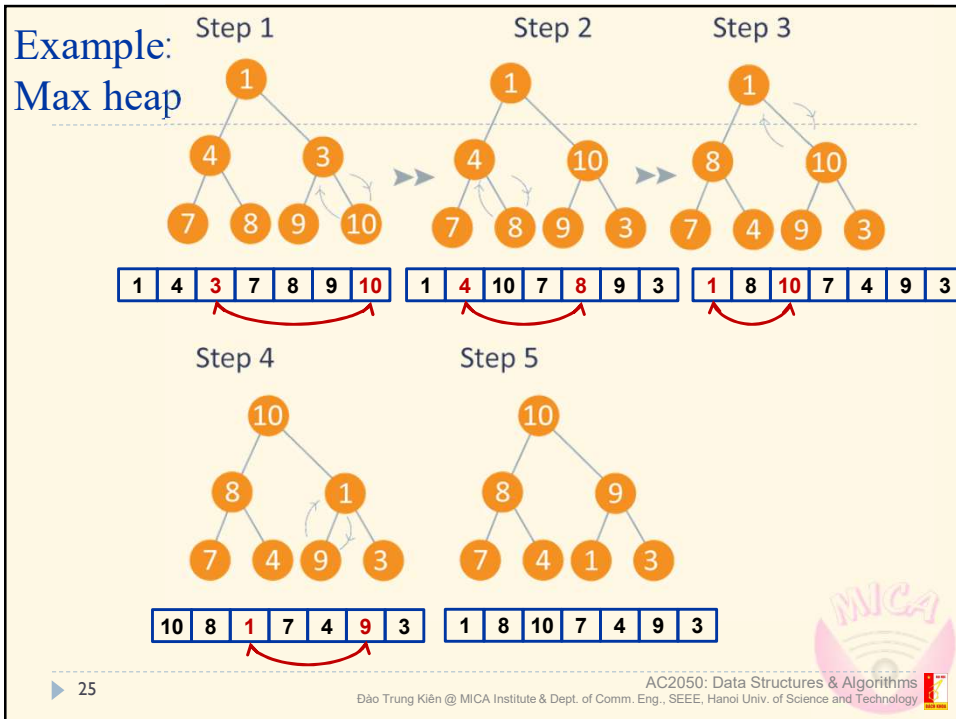


▶ 24

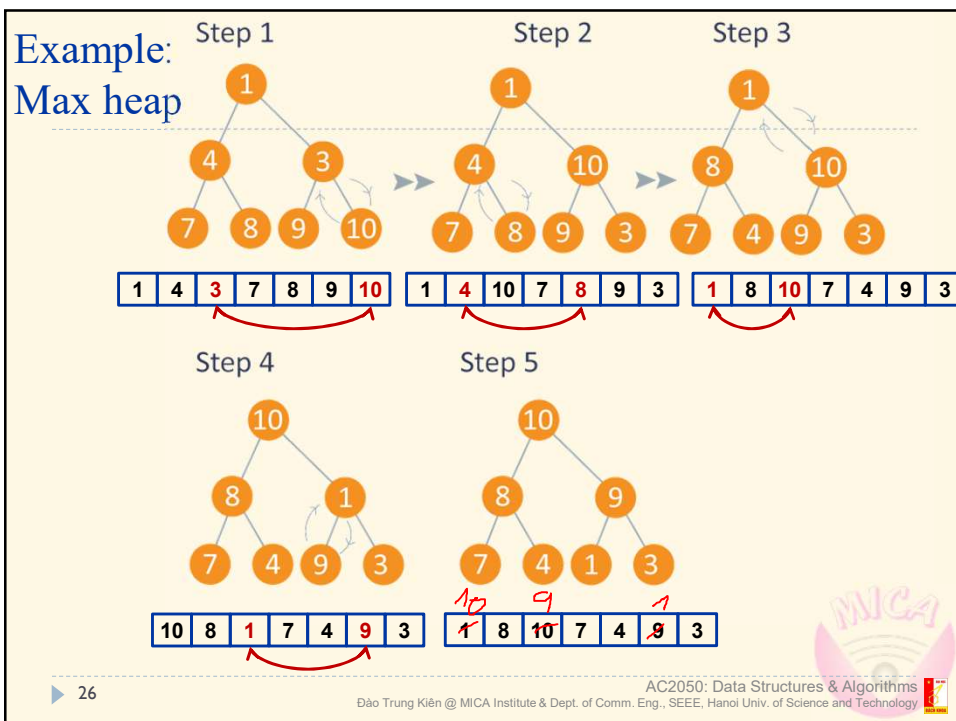
AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



24

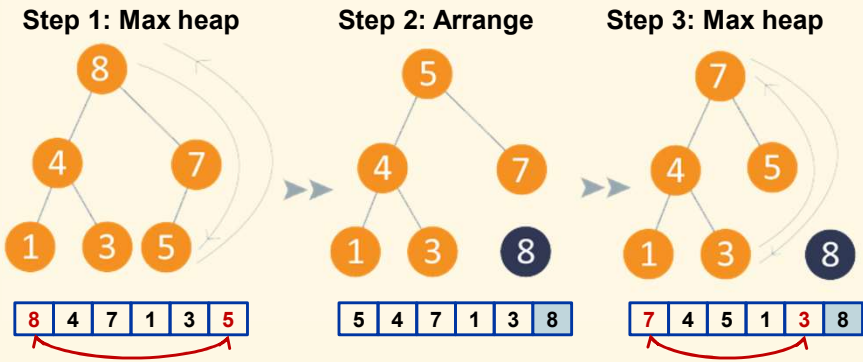


25



26

Example: Heap Sort

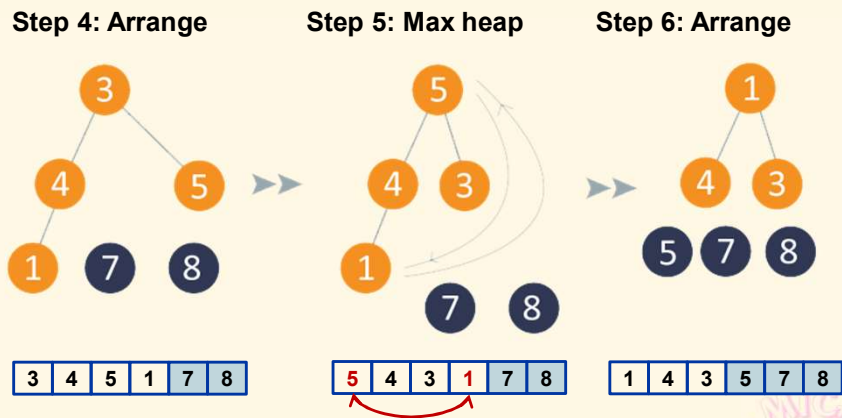


27

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology

27

Example: Heap Sort (cont'd)



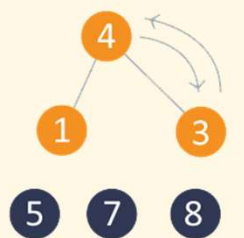
28

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology

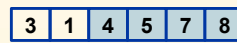
28

Example: Heap Sort (*cont'd*)

Step 7: Max heap



Step 8: Arrange



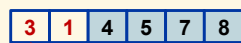
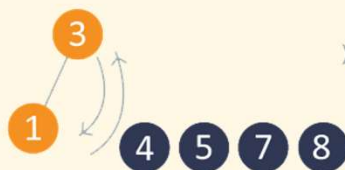
▶ 29

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology

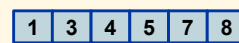
29

Example: Heap Sort (*cont'd*)

Step 9: Max heap



Step 10: Arrange



▶ 30

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology

30

Implementation: Correct Heap Node

```
function correctHeapNode(a, n, i) {  
    const l = i*2 + 1,  
          r = l + 1;  
    let m = i;  
    if (l < n && a[l] > a[m]) m = l;  
    if (r < n && a[r] > a[m]) m = r;  
  
    if (m !== i) {  
        swap(a, i, m);  
        correctHeapNode(a, n, m);  
    }  
}
```

▶ 31

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



31

Implementation: Max Heap

```
function buildHeap(a, n) {  
    for (let i=Math.floor(n/2); i>=0; i--)  
        correctHeapNode(a, n, i);  
}
```

▶ 32

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



32

Implementation: Heap Sort

```
function sortHeap(a) {  
    const n = a.length;  
    buildHeap(a, n);  
  
    for (let i = n; i > 1; i--) {  
        swap(a, 0, i-1);  
        correctHeapNode(a, i-1, 0);  
    }  
}
```

▶ 33

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



33

Complexity

- ▶ Computation
 - ▶ Correct node: $O(\log n)$
 - ▶ Build heap: $O(n \log n)$
 - ▶ Main loop: $O(n \log n)$
 - ▶ Overall: $O(n \log n)$
- ▶ Memory: $O(1)$

▶ 34

AC2050: Data Structures & Algorithms
Đào Trung Kiên @ MICA Institute & Dept. of Comm. Eng., SEEE, Hanoi Univ. of Science and Technology



34