

Week 8

Sorting, HashTables

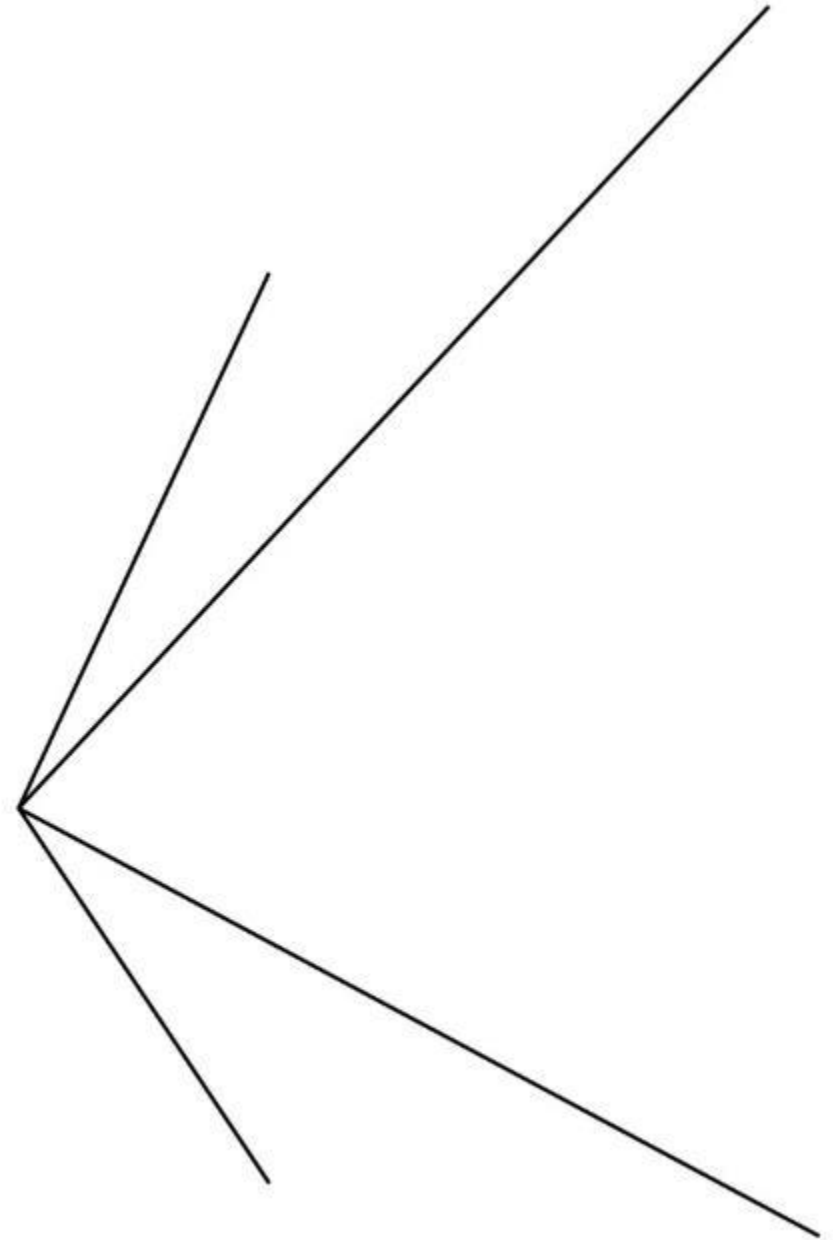
LM Data Structures, Algorithms, and Databases
(34141)

Dr Ahmad Ibrahim
a.ibrahim@bham.ac.uk
March 04, 2024

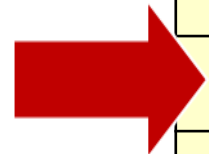


UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Contents based on lecture notes from Uday, Mirco, Martin, Alan



Topics by Week



Week	Date	Topic
1	15 Jan	Searching algorithms
2	22 Jan	Binary Search Tree
3	29 Jan	Balancing Trees – AVL Tree
4	5 Feb	Databases – Conceptual Design
5	14 Feb	Databases – Logical Design & Relational Algebra
6	19 Feb	Consolidation Week
7	26 Feb	Complexity analysis, Stacks, Queues, Heaps
8	4 Mar	Sorting Algorithms, Hash tables
9	11 Mar	Graph Algorithms
10	18 Mar	Databases – Normalization
		Easter break and Eid break
11	22 Apr	Databases – Concurrency
12	29 Apr	Revision Week

Timetable & Office hours

Day	Time	Event	Location
Monday	4:00-5:00pm	Online support session*	Online*
Tuesday	4:00-5:00pm	Office hour 1 (by appointment)*	Online*
Wednesday	-	-	-
Thursday	4:00-5:00pm	Office hour 2 (by appointment)*	Online*
Ramadan Timetable March 17th and March 24th 2024 Sunday 1:00-4:00pm			Auditorium
			Auditorium

Assessments

Assessments (Test 1, Test 2, Test 3): **20%**
Exam: **80%**

Week 10



Test 3

Not available until 20 Mar at 16:00 | Due 21 Mar at 16:00 | -/20 pts

Important Note (Ramadan)

4:00pm UAE Time

(12:00 noon UK Time)



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Weekly Reading

Data Structures and Algorithm Analysis by Clifford A. Shaffer (3rd Ed)



Section 7 Sorting

Section 9.4 Hashing

<https://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf>

Introduction to Algorithms by Cormen (4th Ed)

Section 2.1 Insertion sort

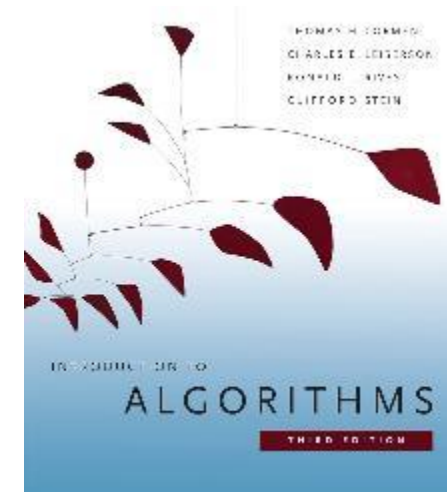
Section 2.3.1 The divide-and-conquer method

Chapter 6 Heapsort

Chapter 7 Quicksort

Chapter 11 HashTables

<https://ebookcentral.proquest.com/lib/bham/detail.action?docID=6925615>



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Last Week

Complexity analysis

Stacks

Queues

Heaps

Sorting Algorithm

Insertion Sort

Heapsort



This Week

Sorting Algorithm

Insertion Sort

Heapsort

Merge sort (Divide and conquer)

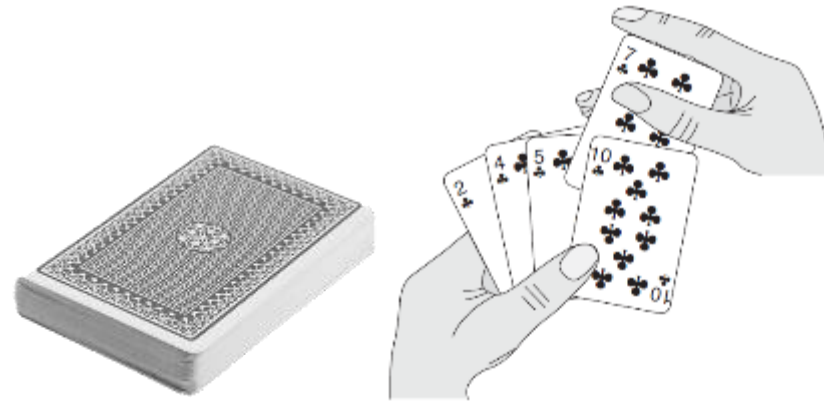
Quick Sort (Divide and conquer)

HashTables



Insertion sort

A simple, intuitive sorting algorithm.



Iteratively removes
one element from
the input data



Finds location it
belongs in the
sorted list

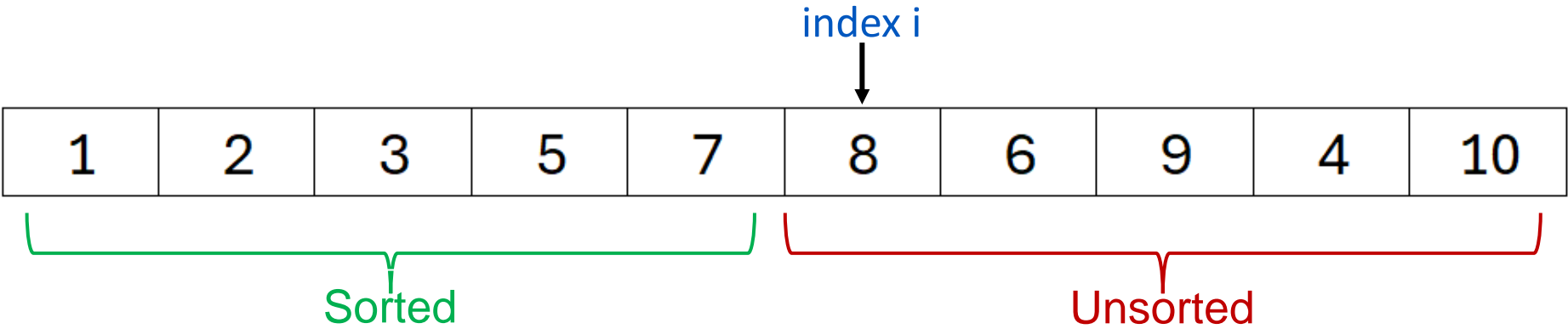
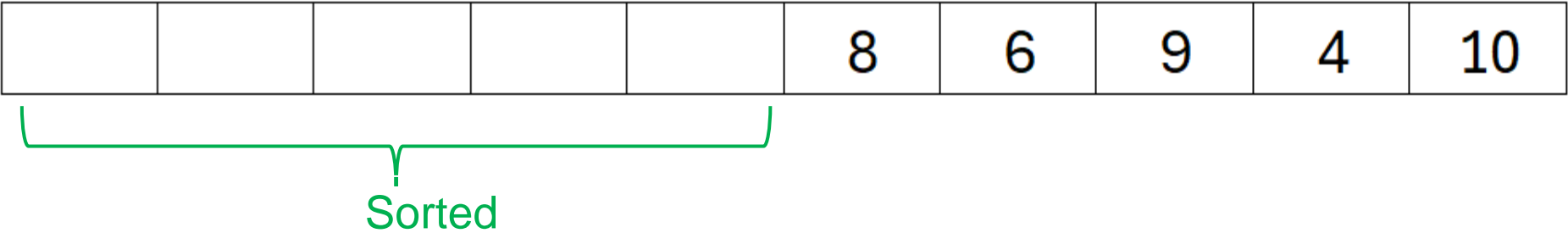
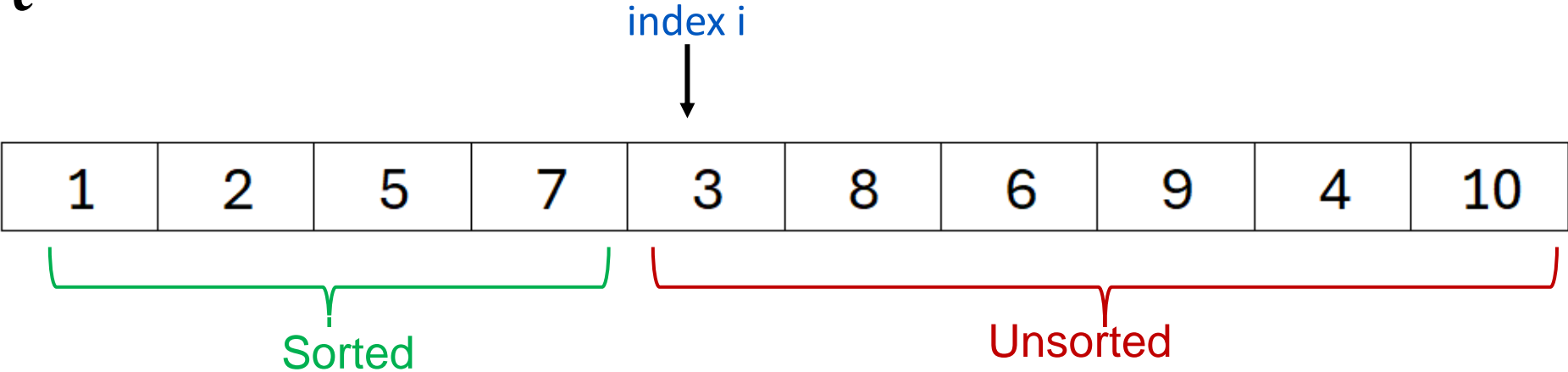


Insert it there until
no input elements
remain

Main steps in Insertion sort.

Insertion Sort

Array (A)



Insertion Sort (Complexity)

Insertion-Sort (A):

```
for i=2 to n
```

```
    Key=A[i]
```

```
    j=i-1
```

```
    while A[j]>Key and j>=0
```

```
        A[j+1] = A[j]
```

```
        j=j-1
```

```
    A[j+1]= Key
```

$O(n)$

$O(n)$

$O(n^2)$



Exercise: Insertion Sort

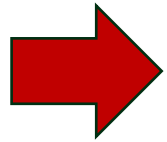
Show step by step working of Insertion sort algorithm for the given array A (5, 2, 4, 6, 1, 3)



This Week

Sorting Algorithm

Insertion Sort



Heapsort

Merge sort (Divide and conquer)

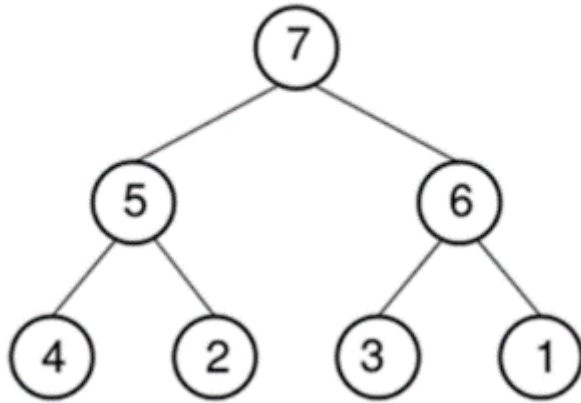
Quick Sort (Divide and conquer)

HashTables



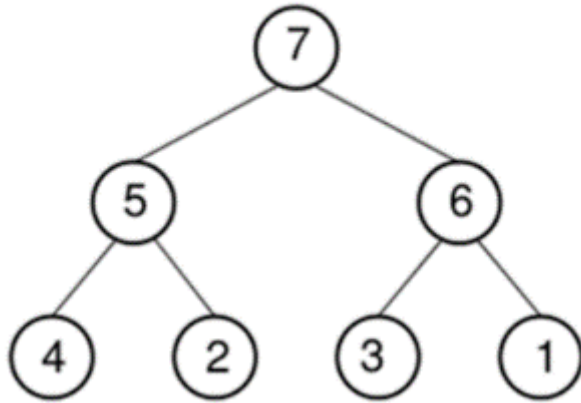
Errata (last week Exercise Sheet 07)

Q6. Show the heap that results from deleting the maximum value from the max-heap of following Figure.

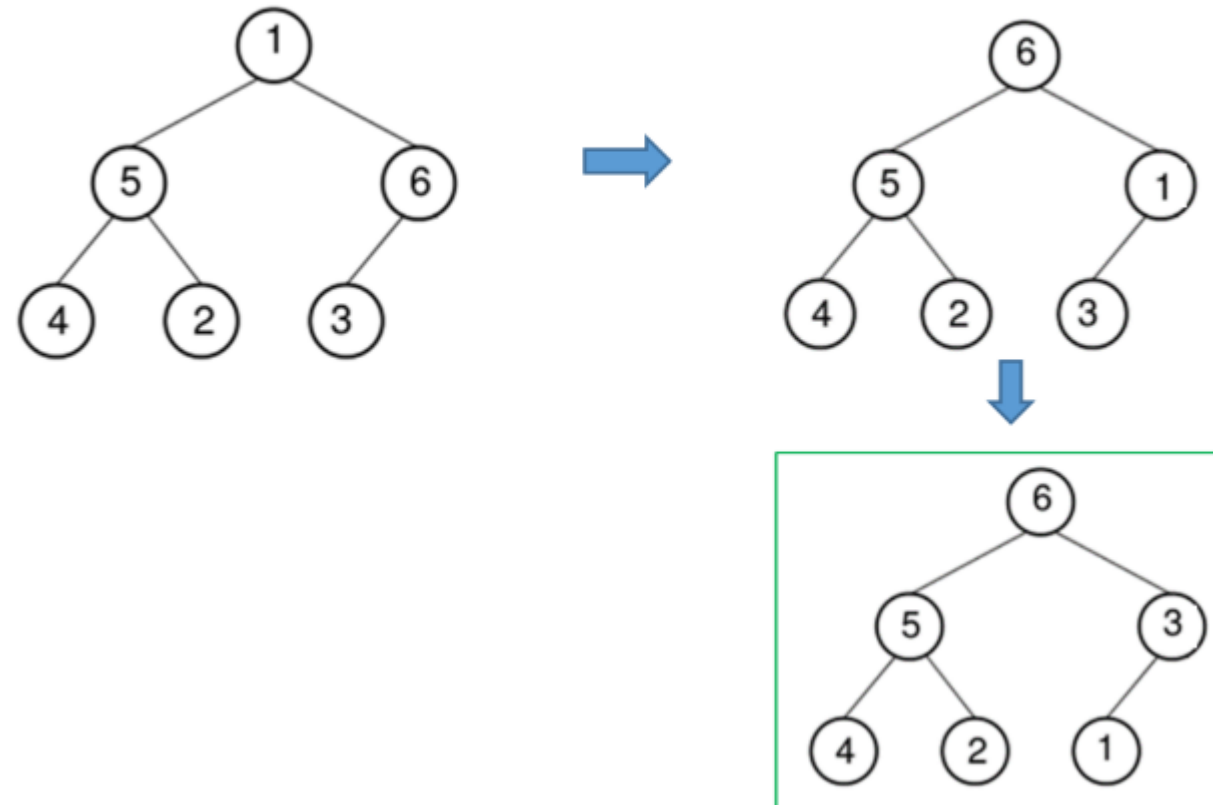


Errata (last week Exercise Sheet 07)

Q6. Show the heap that results from deleting the maximum value from the max-heap of following Figure.



Answer: We can move the element in the last position in the heap (the current last element in the array) to the root position and then order the heap.



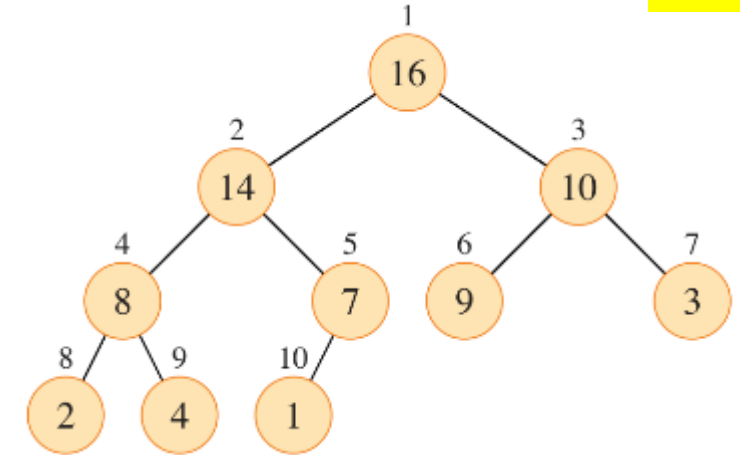
Heapsort

For a Max-Heap, the largest item is stored at the root node.

1.Swap: Remove the root element and put at the end of the array (nth position). Put the last item of the tree (heap) at the vacant place.

2.Remove: Reduce the size of the heap by 1.

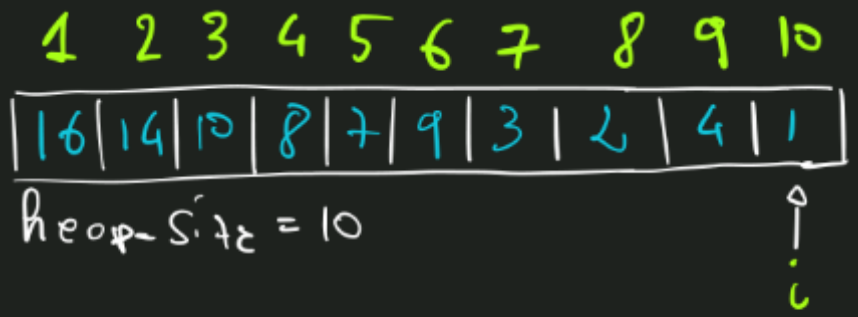
3.Heapify: Heapify the root element again so that we have the highest element at root.



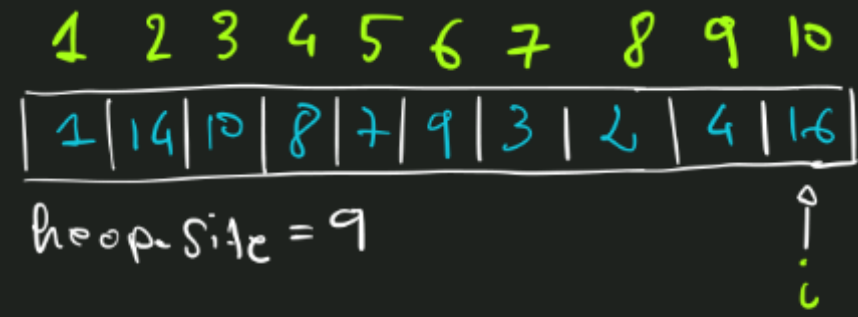
HEAPSORT(A, n)

```

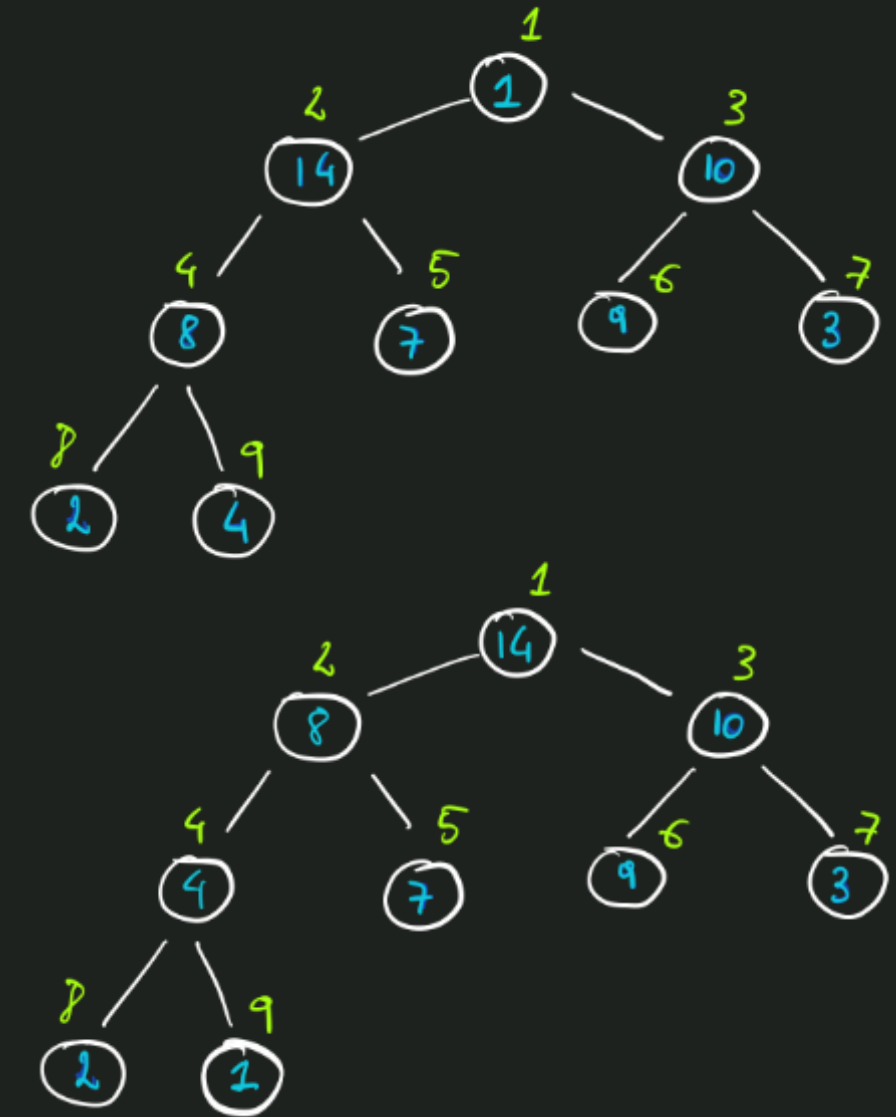
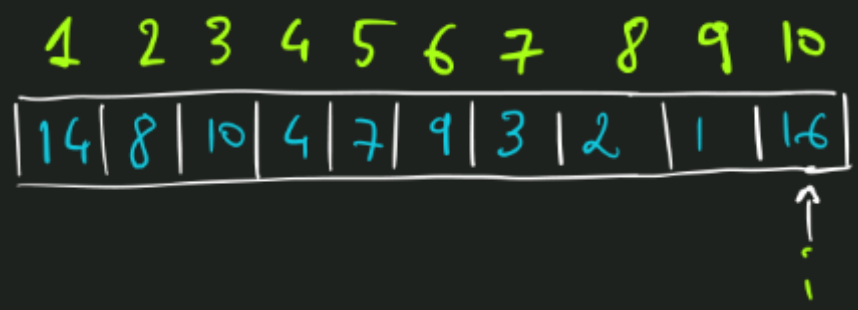
1  BUILD-MAX-HEAP( $A, n$ )
2  for  $i = n$  downto 2
3      exchange  $A[1]$  with  $A[i]$ 
4       $A.heap-size = A.heap-size - 1$ 
5      MAX-HEAPIFY( $A, 1$ )
  
```

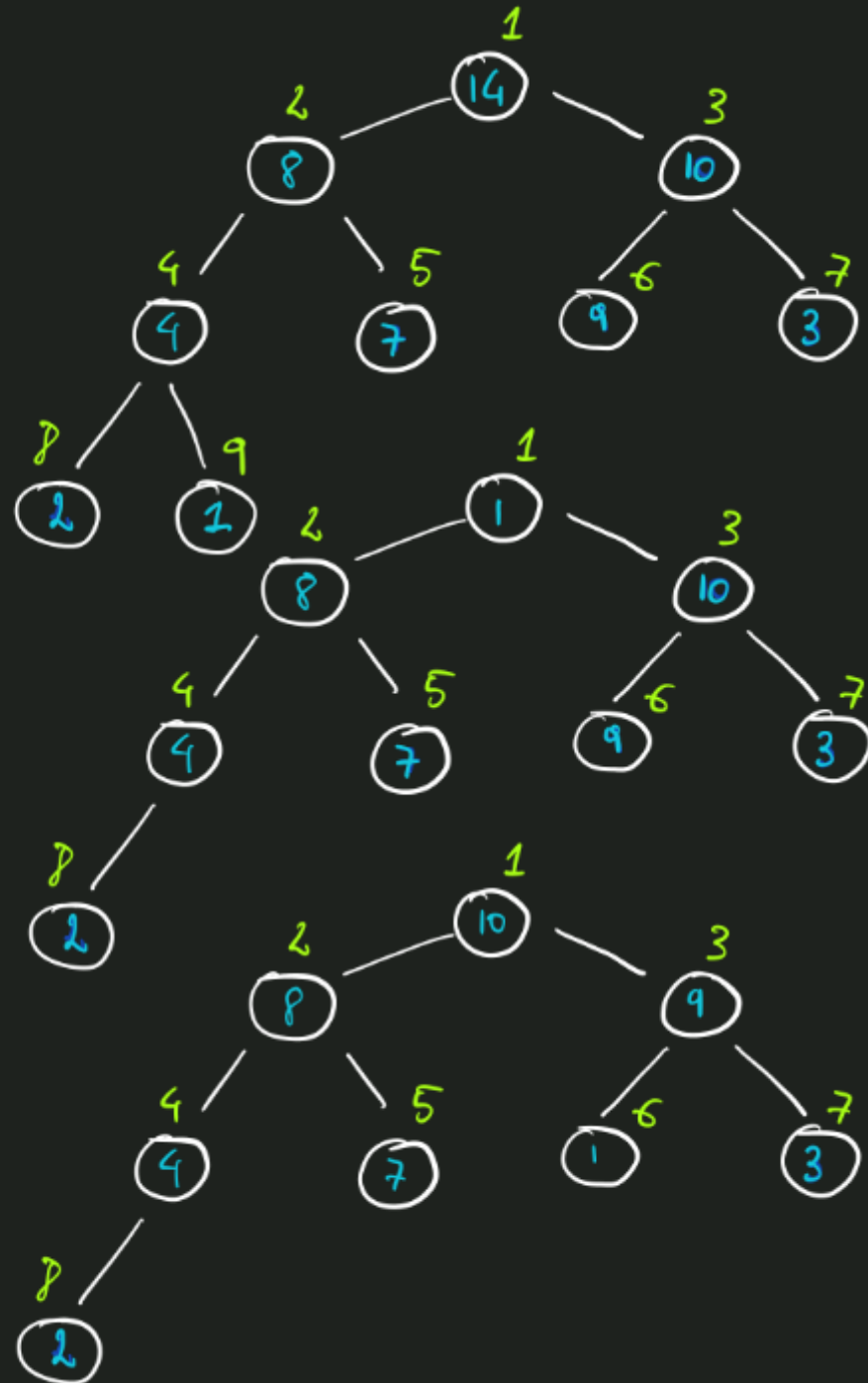
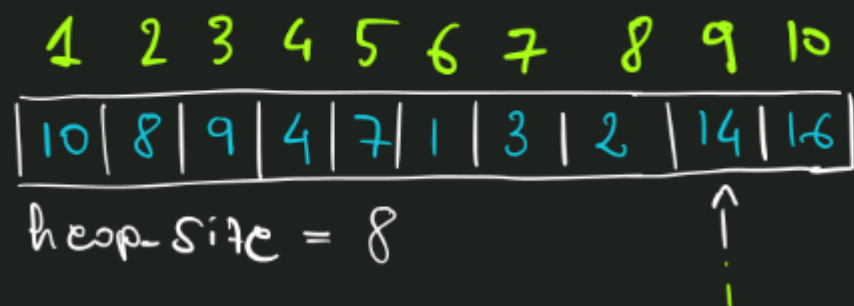
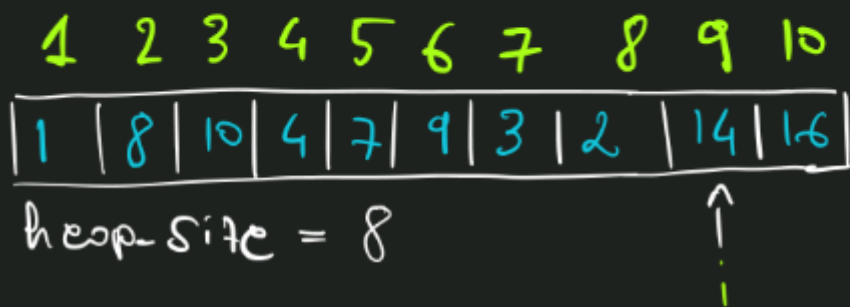
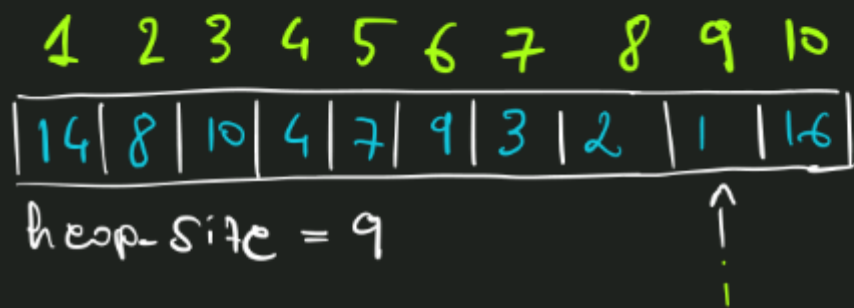



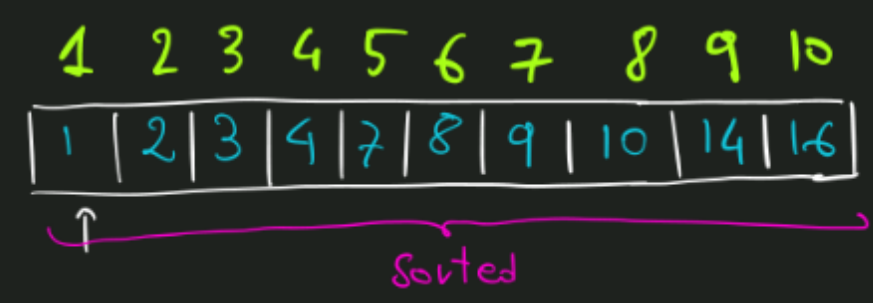
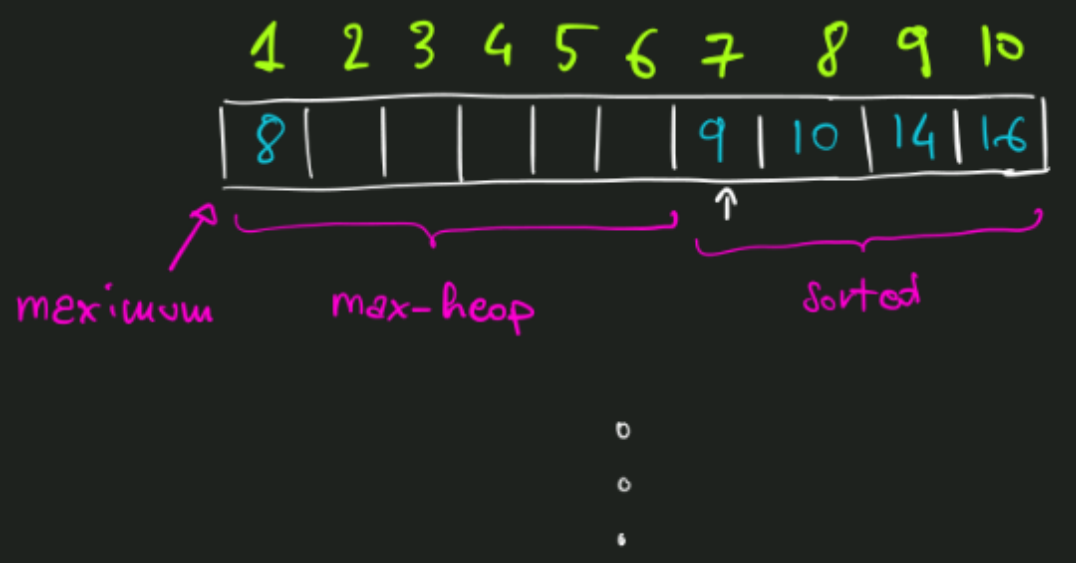
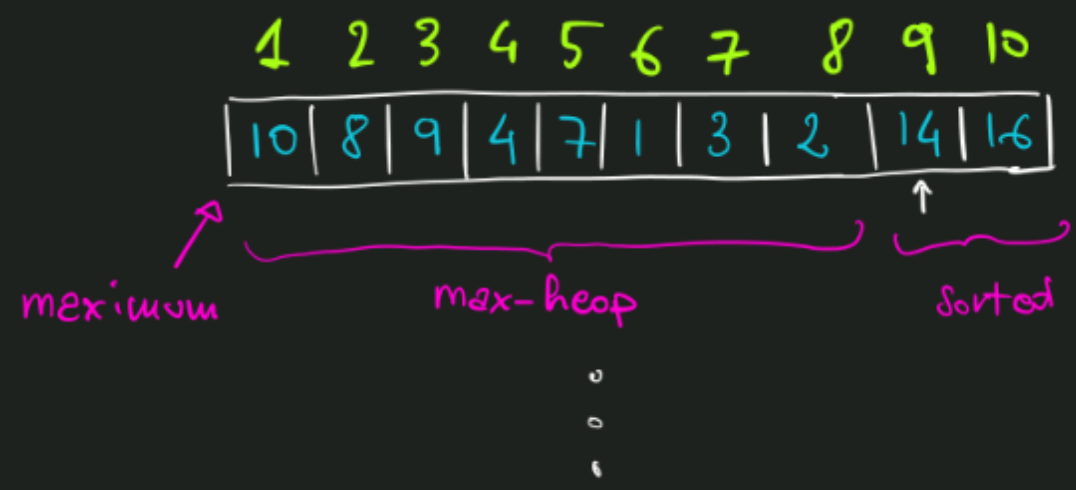
Swap



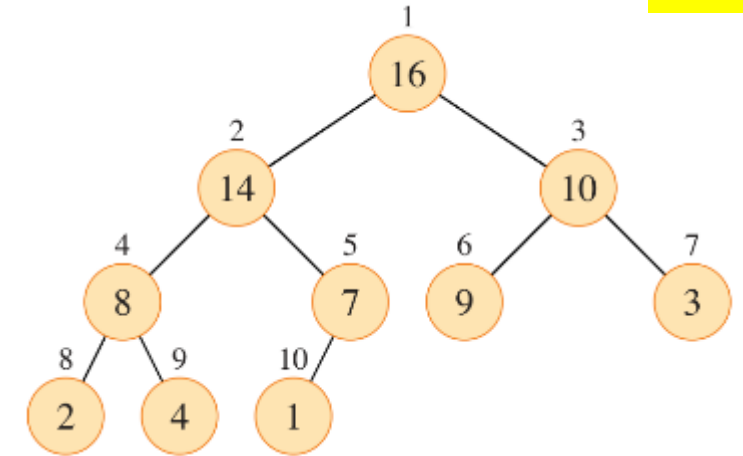
Max-Heapify(A, 1)







Heapsort (Complexity)



HEAPSORT(A, n)

```

1  BUILD-MAX-HEAP( $A, n$ )  →  $O(n)$ 
2  for  $i = n$  downto 2
3      exchange  $A[1]$  with  $A[i]$ 
4       $A.heap-size = A.heap-size - 1$ 
5      MAX-HEAPIFY( $A, 1$ )  →  $O(\log n)$ 
  
```

Exercise Heapsort



Consider the following array

Index	1	2	3	4
Value	9	7	5	11

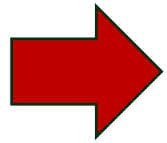
Carry out the steps of sorting in ascending order using HeapSort.

This Week

Sorting Algorithm

Insertion Sort

Heapsort



Merge sort (Divide and conquer)

Quick Sort (Divide and conquer)

HashTables



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Merge sort (Divide and conquer)

Merge Sort (Divide & Conquer)

(Slides from Alan P. Sexton)

Merge-sort.pdf



UNIVERSITY OF
BIRMINGHAM

DUBAI
دبي

Exercise Merge Sort



Consider the following array

9	7	5	11
---	---	---	----

Carry out the steps of sorting in ascending order using Merge Sort.

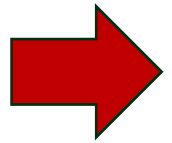
This Week

Sorting Algorithm

Insertion Sort

Heapsort

Merge sort (Divide and conquer)



Quick Sort (Divide and conquer)

HashTables



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Quick sort (Divide and conquer)

Quick Sort (Divide & Conquer)

(Slides from Alan P. Sexton)

Quick-sort.pdf



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Exercise Quick Sort



Consider the following array

9	7	5	11
---	---	---	----

Carry out the steps of sorting in ascending order using quick Sort.

This Week

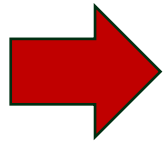
Sorting Algorithm

Insertion Sort

Heapsort

Merge sort (Divide and conquer)

Quick Sort (Divide and conquer)



HashTables



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

HashTables

Hash tables



Exercise HashTable



Create Hashtable for the given input: Input: [120, 130, 241, 253, 367, 446]

Hash Function: $\text{studentID} \bmod 10$

Each student ID is a three-digit number, ranging from 000 to 999.

Summary

Sorting Algorithm

Insertion Sort

Heapsort

Merge sort (Divide and conquer)

Quick Sort (Divide and conquer)

HashTables



Weekly Reading

Data Structures and Algorithm Analysis by Clifford A. Shaffer (3rd Ed)



Section 7 Sorting

Section 9.4 Hashing

<https://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf>

Introduction to Algorithms by Cormen (4th Ed)

Section 2.1 Insertion sort

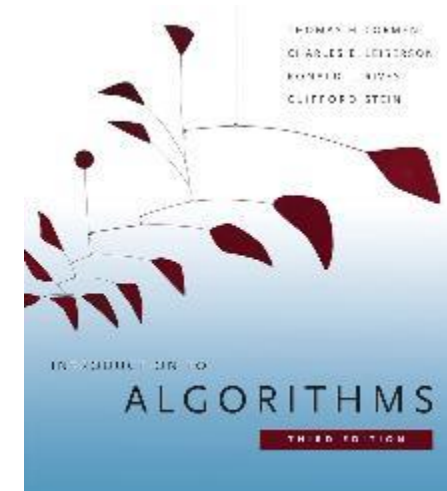
Section 2.3.1 The divide-and-conquer method

Chapter 6 Heapsort

Chapter 7 Quicksort

Chapter 11 HashTables

<https://ebookcentral.proquest.com/lib/bham/detail.action?docID=6925615>



UNIVERSITY OF
BIRMINGHAM | DUBAI
دبي

Next
Week

Week	Date	Topic
1	15 Jan	Searching algorithms
2	22 Jan	Binary Search Tree
3	29 Jan	Balancing Trees – AVL Tree
4	5 Feb	Databases – Conceptual Design
5	14 Feb	Databases – Logical Design & Relational Algebra
6	19 Feb	Consolidation Week
7	26 Feb	Complexity analysis, Stacks, Queues, Heaps
8	4 Mar	Sorting Algorithms, Hash tables
9	11 Mar	Graph Algorithms
10	18 Mar	Databases – Normalization
		Easter break and Eid break
11	22 Apr	Databases – Concurrency
12	29 Apr	Revision Week



Thank you.
Questions?