College of Science and Computer Engineering, Yanbu

TAIBAH UNIVERSITY



CS211 Algorithms & Data Structures

Lecture 6

1443 - 2021

Dr. Sameer Mabrouk Alrehaili
College of Science and Computer Engineering, Yanbu

Chapter 7

Sorting Algorithms

Objectives

• One of common way to teach the complexity of algorithm is to take sorting problem as an example.

•

Objectives

- Bubble sort
- Selection sort
- Insertion sort

•

Objectives

 Sorting algorithm receives a set of objects as input, and arranges this set of elements either in ascending or descending order.



Algorithms & Data Structures CS 211 Objectives

- Objective
 - To arrange a list of elements in some order
- Unsorted array

C

Sorted in ascending order

0

Sorted in descending order

0

Growth rates

Function	Name	
С	constant	
Log N	Logarithmic	
Log N	Log-squared	
N	Linear	
N log N		
N2	Quadratic	
N3	Cubic	
2N	Exponential	

V	7						
0	1	2	3	4	5	6	
2	8	5	3	9	4	1	No swap
		1					
0	1	2	3	4	5	6	
2	8	5	3	9	4	1	swap
			1				
0	1	2	3	4	5	6	
2	5	8	3	9	4	1	swap
			√	7			
0	1	2	3	4	5	6	_
2	5	3	8	9	4	1	No swap

					7		
0	1	2	3	4	5	6	_
2	5	3	8	9	4	1	swap
						1	
0	1	2	3	4	5	6	
2	5	3	8	4	9	1	swap

	F	7						
	0	1	2	3	4	5	6	
	2	5	3	8	4	1	9	No swap
			1					
_	0	1	2	3	4	5	6	
	2	5	3	8	4	1	9	swap
	0	1	2	3	4	5	6	
	2	3	5	8	4	1	9	No swap
				6				
	0	1	2	3	4	5	6	_
	2	3	5	8	4	1	9	wap

		1					
_	6	5	4	3	2	1	0
swap	9	1	8	4	5	3	2

V	7						
0	1	2	3	4	5	6	
2	3	5	4	1	8	9	No swap
		1					
0	1	2	3	4	5	6	
2	3	5	4	1	8	9	No swap
			7				
0	1	2	3	4	5	6	
2	3	5	4	1	8	9	swap
			<u> </u>	7			
0	1	2	3	4	5	6	_
2	3	4	5	1	8	9	swap

	1						
0	1	2	3	4	5	6	
2	3	4	1	5	8	9	No swap
		7					
0	1	2	3	4	5	6	_
2	3	4	1	5	8	9	No swap
		F	7				
0	1	2	3	4	5	6	_
2	3	4	1	5	8	9	swap

	A						
0	1	2	3	4	5	6	_
2	3	1	4	5	8	9	No swap
		1					
0	1	2	3	4	5	6	
2	3	1	4	5	8	9	swap

Bubble sort

V	7						
0	1	2	3	4	5	6	
2	1	3	4	5	8	9	swap
	4	0	0	4	_	0	

3

Bubble sort

One of the simplest sorting algorithm is called bubble sort. The idea is to compare two consecutive items, swap them if they are in reverse order, and repeat.

```
Input: An array A of n numbers A \leftarrow \{x_1, \dots, x_n\}
Output: sorted A
1: for i \leftarrow 1 to n-1 do
2: for j \leftarrow 2 to n-i do
3: if A_{j-1} > A_j then
4: swap A_{j-1} and A_j
5: end if
6: end for
7: end for
```