CS211
Algorithms & Data Structures

Lecture 1

Fall 1443 - 2021
Dr. Sameer Mabrouk Alrehaili
College of Science and Computer Engineering, Yanbu

Chapter 1

INTRODUCTION

Algorithms & Data Structures CS 211

Objectives

- A brief review of programming concepts and discrete mathematics.
- History of Algorithms.
- What are algorithms?
- What are data structures?
- Briefly review recursion.
- The selection problem
- Word puzzle

Algorithms & Data Structures CS 211

History of Algorithms

- Did you know that “**Algorithms**” were used and implemented when there were no computers?
- **Algorithms** were named after a great muslim mathematician, Muhammad ibn Musa Al-Khwarizmi, who presented the first systematic solution of linear and quadratic equations in his book. (المختصر في حساب الجبر والمقابلة).
- Al-khwarizmi also known as “The father of Algebra”.

Algorithms & Data Structures CS 211

What are algorithms?

- Imagine you are asked to do some of the following tasks:
  - To give a friend directions to your home.
  - To change a car oil.
  - To make a cupcake.
- Each of above task can be solved as a set of instructions.
- In school, pupils are being taught how to multiply numbers which is a simple algorithm.
- Therefore, they can be called as algorithms.

Algorithms & Data Structures CS 211

An algorithm for preparing an omelette

1. How to make an omelette
2. Get a bowl and a whisk
3. Do you have a whisk?
4. If yes, goto 9
5. If no, find a fork
6. Do you have a fork?
7. If yes, goto 9
8. If no, do not worry, do not beat them
9. Beat the eggs
10. Get a butter
11. Do you have a butter?
12. etc ...

Algorithms & Data Structures CS 211

Need

- Complex algorithms enable us to quickly access information (Search engines such as Google, Yahoo, and Bing).
- Finding good routes to transfer a packet from source to the destination (Networking).
- To learn from data and improve from experience without human intervention (AI).
- Algorithms help you planning your route when you provide your destination.
- Algorithms assist you (Siri, Alexa, Google Assistant, and Cortana)
- Map shortest path from point to another
-

Algorithms & Data Structures CS 211

What are algorithms?

- An algorithm is "**a finite** sequence of instructions, each of which has **a clear meaning** and can be performed with a finite amount of effort in a finite length of time".
- An algorithm is a clearly specified set of simple instructions to be followed to solve a problem.
- An algorithm is a well-defined procedure that allows a computer to solve a problem.
- An algorithm is a sequence of **unambiguous** instructions.

Algorithms & Data Structures CS 211

Computational problem

- The following algorithm Average computes the average of the examination scores Score[] of n students.

  Sorting problem.

  Input : A sequence of n numbers {a1, a2, …, an}
  Output: A permutation (reordering) {a1, a2, …, an} of the input sequence such that a1<a2<...<an

Algorithms & Data Structures CS 211

What are Data Structures?

- In this course, we shall typically describe algorithms as programs written in a pseudocode that is similar in many respects to C, C++, Java, Python, or Pascal.
- The following algorithm Average computes the average of the examination scores Score[] of n students.

Average(S, n)
Input : S[]: array of score data, n: the number of scores
Output: Average score

1. x ← 0;
2. **for** i ← 1, 2, ..., n **do**
3. x ← x + S[i];
4. **end**
5. output x/n;

Algorithms & Data Structures CS 211

What are Data Structures?

- You can solve a problem such as queue and stack using arrays, but it would be more efficiently solved if you use the appropriate data structure.

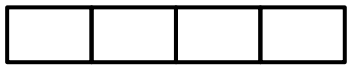
Algorithms & Data Structures CS 211

What are Data Structures?

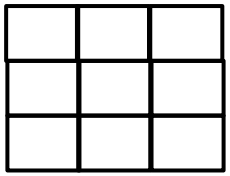
- The term data structure is used to denote a particular way of organising data for particular types of operation.
- For example to calculate the area of a circle you need to store values of double of float. While if you want to
- A particular way of storing and organising data in a computer so that it can be used efficiently.
- The choice of data structure is based on which type of operation is required.
- Wrong choice of data structure to solve a particular problem will affect the performance of solution.
- Simply, data structure is different ways of storing data.

Algorithms & Data Structures CS 211

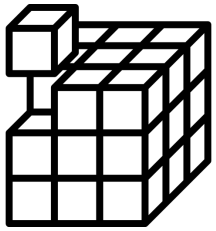
Examples of Data Structures



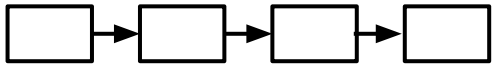
1D array



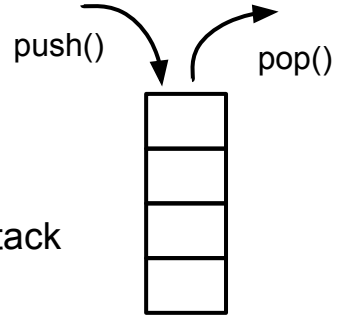
2D array (3X3)

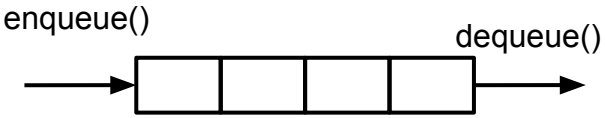


ND array (3X3X3)

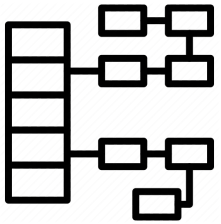


Linked List

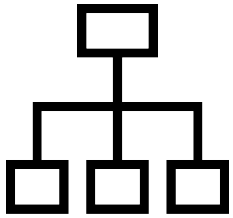




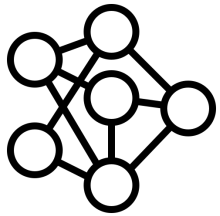
Queue



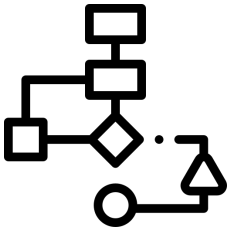
Hash



Tree



Graph



Algorithms & Data Structures CS 211

The selection problem

- Suppose you have a set of N numbers and would like to determine the kth largest, (This is known as the selection problem).
- solution1
  - Read N numbers into an array
  - Sort the array in descending order
  - Return the element in position k
- Solution2
  - Read the first k elements into an array
  - Sort them in decreasing order
  - Each remaining element is read one by one
  - If it is smaller then kth element in the array ignore
  - Otherwise, it is placed in its correct spot in the array
- Which algorithm is better?
- Each requires several days of computer processing to terminate
- Neither algorithm finishes in a reasonable amount of time if k =15,000,000
- They work, but can not be considered good algorithms
- Impractical

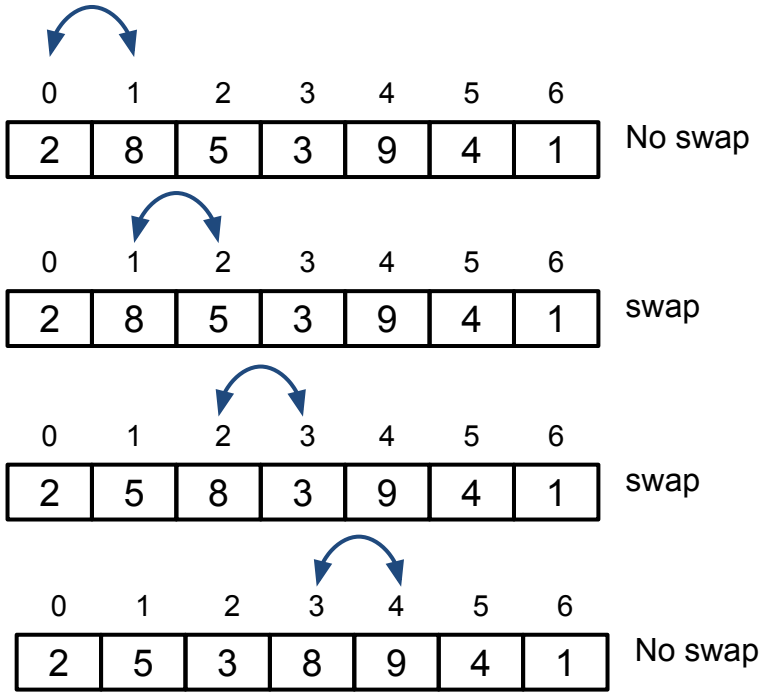
Algorithms & Data Structures CS 211

Word puzzle

- You have 2d array of letters and a list of words. The object is to find the words in the puzzle. These words may be horizontal, verical, or diagonal in any direction. Example (1,1) to (1,4) **this**, (1,1) to (3,1) **two**, and (4,1) to (2,3) **fat**.
- Solution 1
  - For each word in the word list, we check each ordered triple (row, column, orientation) for the presence of the word. This amounts to lots of nested for loops but is basically straightforward.
- Solution 2
  - for each ordered quadruple (row, column, orientation, number of characters) that doesn't run off an end of the puzzle, we can test whether the word indicated is in the word list. Again, this amounts to lots of nested for loops. It is possible to save some time if the maximum number of characters in any word is known.

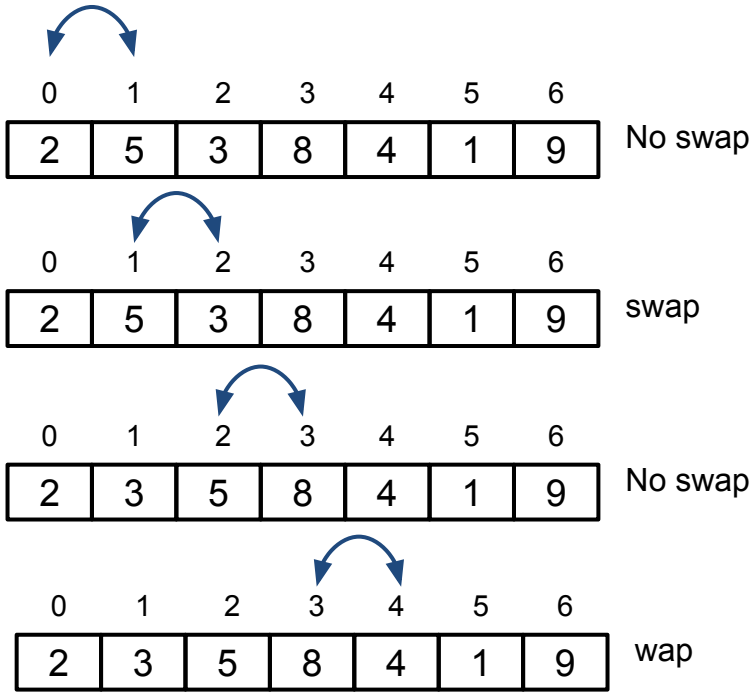
Algorithms & Data Structures CS 211

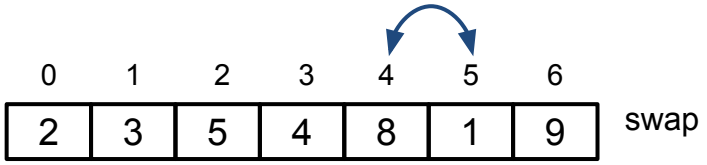
Bubble sort



Algorithms & Data Structures CS 211

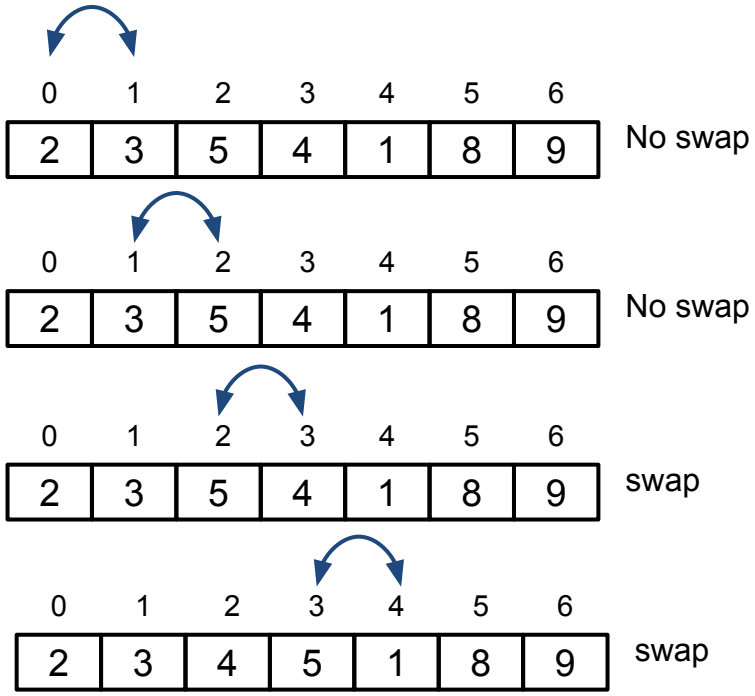
Bubble sort





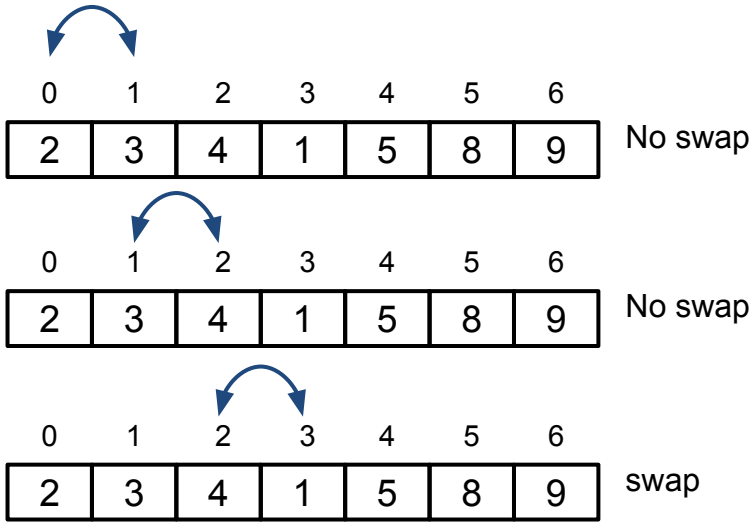
Algorithms & Data Structures CS 211

Bubble sort



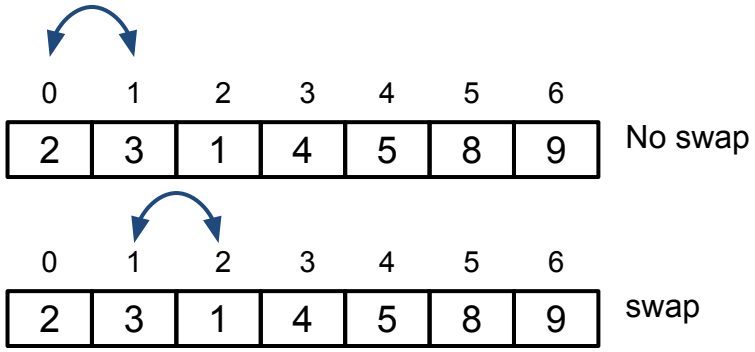
Algorithms & Data Structures CS 211

Bubble sort



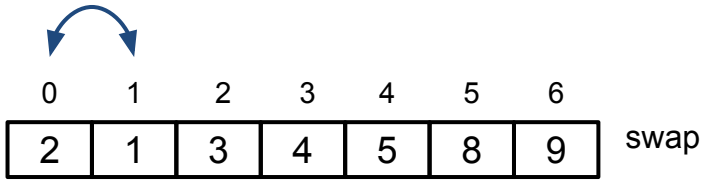
Algorithms & Data Structures CS 211

Bubble sort



Algorithms & Data Structures CS 211

Bubble sort



Algorithms & Data Structures CS 211

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.

```
    BubbleSort(a)
    Input : An array a[] of size n
    Output: An array a[] that is sorted
 1. for j ← n-1, n-2, …, 2 do
 2.         for i ←1, 2, …, j do
 3.                 if a[i] > a[i+1] then
 4.                         tmp ← a[i];
 5.                         a[i] ← a[i+1];
 6.                         a[i+1] ← tmp;
 7.                 end
 8.         end
 9. end
10. output a[];
```

Algorithms & Data Structures CS 211

Assignment

- sfs