CS 2009
Design and Analysis of Algorithms

Prepared by: Waheed Ahmed

Edited by: Farrukh Salim Shaikh
Email : farrukh.salim@nu.edu.pk

Lecture 1:

Introduction & Course Overview

Grading Policy (CS 2009)

Assessment Type	**Weight**
Assignments	10
Midterms (Week 6 & Week 11)	30 (15 each)
Project	10
Final	50

Text & Reference Books

- Required Textbook
  - Thomas H. Cormen “Introduction to Algorithms” 2nd Edition

- Reference Books
- Anany Levitin “Introduction to the Design and Analysis of Algorithms” 3rd edition
- Jon Kleinberg and Éva Tardos “Algorithm Design”
- Sanjoy Dasgupta et al. “Algorithms”
- Steven S. Skiena “The Algorithm Design Manual”

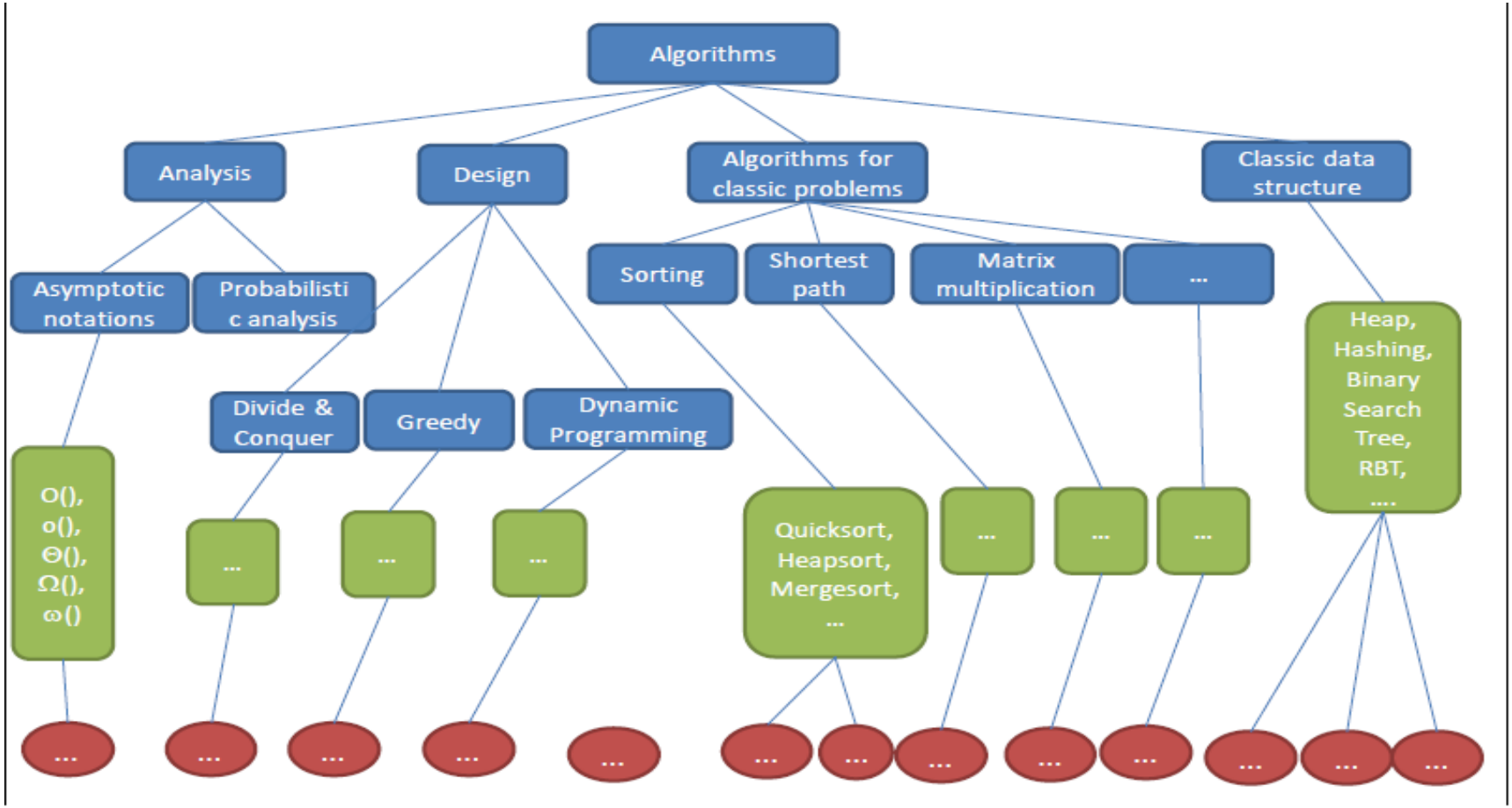
Contents & Tentative Schedule

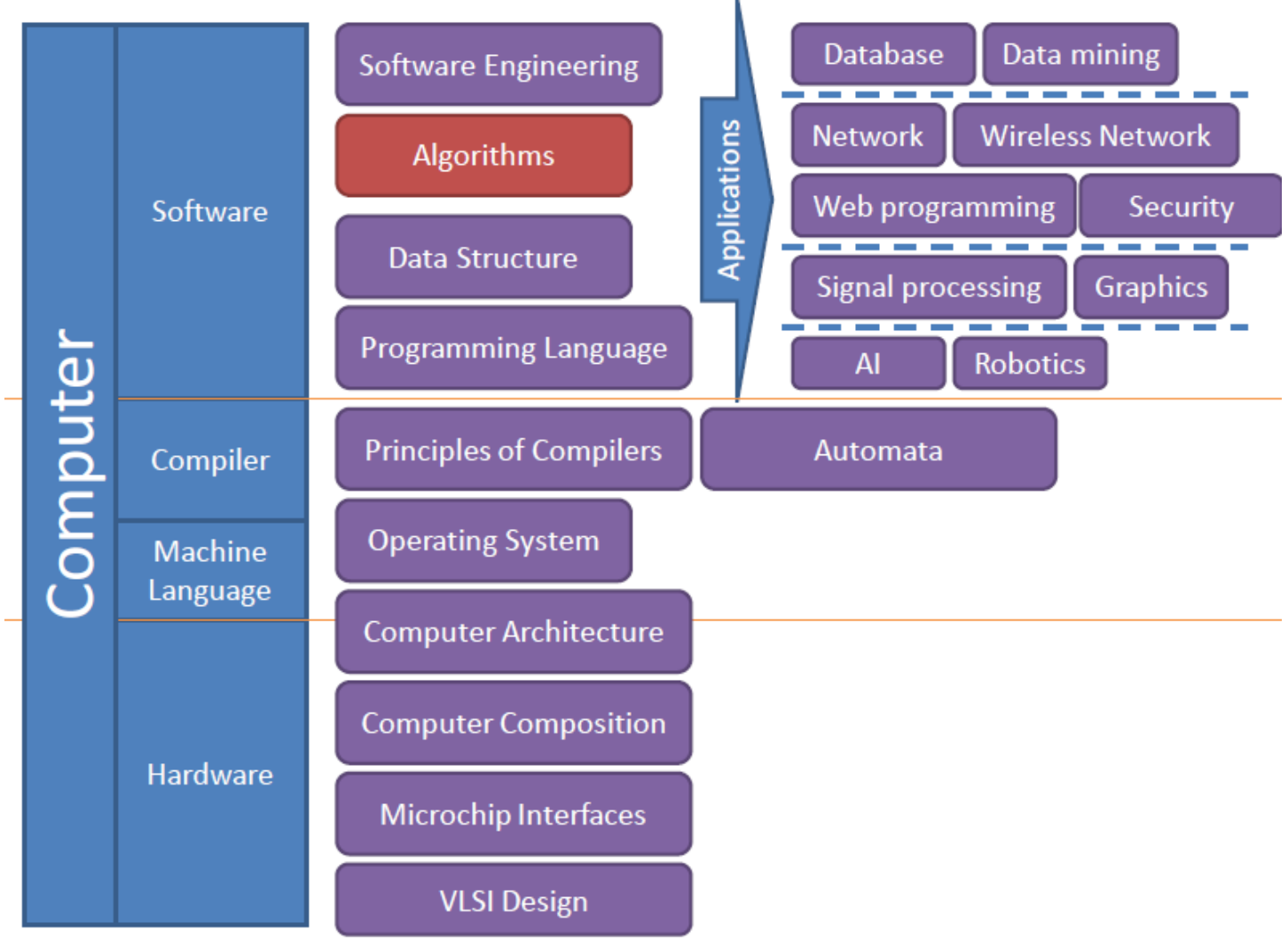
Week	Topics
Week 1 & 2	Basics of Algorithms, Mathematical Foundation, Growth of Function, Asymptotic Notations. Data Structures Review (Stack, Queue, Linked List, Hash Table, Binary Tree).
Week 3 & 4	Divide and Conquer, Substitution Method, Recurrence-Tree Method, Master's Method.
Week 5	Sorting (Merge, Insertion, Quick, Heap, Counting, Radix, Bucket)
Week 6	Mid term 1 Exam
Week 7	Dynamic Programming

Contents & Tentative Schedule

Week	Topics
Week 8	Dynamic Programming & Greedy Algorithms
Week 9, 10 & 12	Graph Theory (Graph Categorization, Graph Terminology, Representation of Graphs, BFS & DFS, Strongly Connected Components, Greedy Algorithms: Kruskal's Algorithm, Prim's Algorithms, Bellman-Ford Algorithms, Dijkstra's Algorithm)
Week 11	Midterm 2
Week 13 & 14	Geometric Algorithms (Introduction,Graham Scan, Close Points). String Matching
Week 15 & 16	NP Complete Problems and Solutions using Approximation Algorithm, Amortized algorithms
Week 17	Review & Project Presentations

Knowledge tree

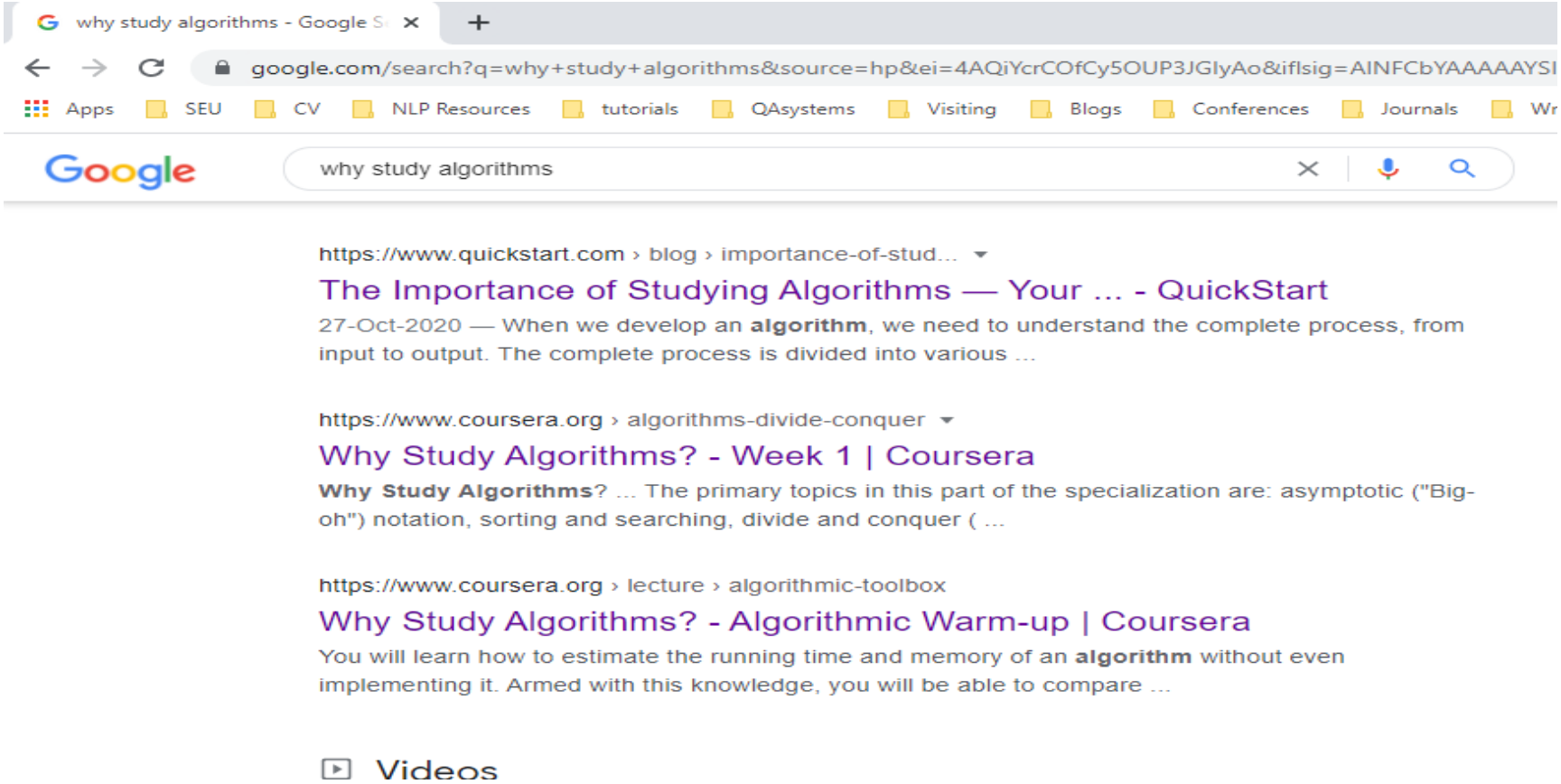



Computer
Software
Compiler
Machine Language
Hardware
Software Engineering
Algorithms
Data Structure
Programming Language
Applications
Database
Data mining
Network
Wireless Network
Web programming
Security
Signal processing
Graphics
AI
Robotics
Principles of Compilers
Automata
Operating System
Computer Architecture
Computer Composition
Microchip Interfaces
VLSI Design

What is an algorithm?

Why Study Algorithms ?

Web Search


why study algorithms - Google S
google.com/search?q=why+study+algorithms&source=hp&ei=4AQiYcrCOfCy5OUP3JGIyAo&iflsig=AINFCbYAAAAAYSI
Apps
SEU
CV
NLP Resources
tutorials
QAsystems
Visiting
Blogs
Conferences
Journals
Google
why study algorithms
https://www.quickstart.com › blog › importance-of-stud...
The Importance of Studying Algorithms — Your ... - QuickStart
27-Oct-2020 — When we develop an algorithm, we need to understand the complete process, from input to output. The complete process is divided into various ...
https://www.coursera.org › algorithms-divide-conquer
Why Study Algorithms? - Week 1 | Coursera
Why Study Algorithms? ... The primary topics in this part of the specialization are: asymptotic ("Big-oh") notation, sorting and searching, divide and conquer (...
https://www.coursera.org › lecture › algorithmic-toolbox
Why Study Algorithms? - Algorithmic Warm-up | Coursera
You will learn how to estimate the running time and memory of an algorithm without even implementing it. Armed with this knowledge, you will be able to compare ...
Videos

Personalized Recommendation

- More than **70% of what people watch** on YouTube is determined by its **recommendation algorithm**.

- News Feed, Friend Suggestions

Human Genome Project

Security E-commerce

Lot of Applications

Internet. Web search, Packet routing, distributed file sharing,...

Biology. Human genome project, protein folding, …

Data Mining. Text Classification, Text Clustering, Page Rank

Security. E-commerce, Cell phones, Voting machine

Web programing. Sorting Algorithms, Searching algorithms

Graphics. Video Games, Virtual Reality,

Social networks. Recommendations, news feed

Machine Learning AI. Linear Regression Algorithm, Deep Neural Networks such RNN, CNN

Robotics. Planning Algorithms.

Why Study Algorithms?

- To become proficient programmer.
- To solve problems that could not be solved.
- For fun and profit.

What is Algorithm

- An algorithm is any well-defined computational procedure that takes some value as input and produces some value as output. (Thomas H. CORMEN)

- An ***algorithm*** is a sequence of **computational** steps for solving a problem.

  E.g.

  - Multiply Two Numbers.
  - Algorithms to Sort Array.

What’s more important than performance?

- Correctness
- Robustness
- User-friendliness
- Simplicity
- Extensibility
- Reliability

MULTIPLICATION PROBLEM

Input: 2 numbers, x and y (n digits each)

Output: the product x · y

$$\begin{array}{r} 2143 \\ \times\ 9112 \\ \hline 19427016 \end{array}$$

MULTIPLICATION PROBLEM

How efficient is this algorithm?

(How many single-digit operations are required?)

Algorithm description (informal*):

compute partial products (using multiplication & “carries” for digit overflows), and add all (properly shifted) partial products together

```
   2143
x  9112
------
   4286
  21430
 214300
19187000
--------
19427016
```

MULTIPLICATION PROBLEM

How efficient is this algorithm?

(How many single-digit operations are required?)

n partial products: ~$2n^2$ ops (at most n multiplications & n additions per partial product)

adding n partial products: ~$2n^2$ ops

(a bunch of additions & "carries")

~ $4n^2$ operations in the worst case

```
   2143
x  9112
-------
   4286
  21430
 214300
19187000
-------
19427016
```

What we are interested in Algorithms

- Correctness
  - Does it work correctly?
- Performance/Efficiency
  - How much time will it take? (Time Complexity)
  - How much space will it take? (Space Complexity)
- Can We do it better?