CS141: Intermediate Data Structures and Algorithms

Introduction

Instructor: Amr Magdy
TAs: Mayur Patil and Tanmay Shah
Computer Science and Engineering
Welcome to CS 141

Instructor: Amr Magdy
Office: Tomas Rivera Library, 159B
http://www.cs.ucr.edu/~amr/
Email: amr@cs.ucr.edu
(Include [CS141] in the subject)
Office hours: MW: 2:00 - 4:00 PM

TAs:

› Mayur Patil
   Office hours: Tuesday: 9:00 – 11:00 AM
   Email: mpati005@ucr.edu
   Office: Chung Hall, room 110

› Tanmay Shah
   Office hours: Tuesday: 2:00 – 3:00 PM
   Email: tshah015@ucr.edu
   Office: Chung Hall, room 110
   (Include [CS141] in the subject)

Course Website: https://www.cs.ucr.edu/~amr/#teaching
Anonymous Feedback Form

Fall 2019: CS 141 - Intermediate Data Structures and Algorithms
Fall 2018: CS 141 - Intermediate Data Structures and Algorithms
Spring 2018: CS 260 - Spatial Data Modeling and Analysis
Winter 2018: CS 141 - Intermediate Data Structures and Algorithms
Anonymous Feedback Form

CS 141 - Intermediate Data Structures and Algorithms

Time: Section 001: M W F - 1:00 PM to 1:50 PM
Section 002: M W F - 5:00 PM to 5:50 PM

Location: Section 001: Winston Chung Hall - Room 143
Section 002: Winston Chung Hall - Room 143

Instructor: Amr Magdy - amr@cs.ucr.edu - Office Hours: 159B Tomas Rivera Library - Monday & Wednesday 2:00 PM - 4:00 PM
TA: Mayur Patil - Office Hours: WCH 110 WCH - Thursday 9:00 - 11:00 AM
Tanmay Shah - Office Hours: 110 WCH - Tuesday 2:00 - 3:00 PM


Click HERE for Teaching Feedback form
Introduction to Computational Algorithms
Computer Programs

- Algorithms + Data Structures = Programs
  - By Niklaus Wirth, Turing award winner 1984

Note: this is not the course textbook. The textbook is provided later.
What is Algorithm?

https://www.youtube.com/watch?v=oRkNaF0QvnI
What is Algorithm?

- According to Merriam-Webster dictionary
  - a procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation;
  - *broadly*: a step-by-step procedure for solving a problem or accomplishing some end especially by a computer.
What is Algorithm?

- According to Merriam-Webster dictionary
  - a procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation; 
  - *broadly*: a step-by-step procedure for solving a problem or accomplishing some end especially by a computer.

- The word originates from “Algorism”: a mathematical counting technique
  - Stemmed from the name of “Muhammad ibn Musa al-Khwarizmi”, an influencer mathematician
What is Algorithm?

- According to Merriam-Webster dictionary:
  - a procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation; 
  - broadly: a step-by-step procedure for solving a problem or accomplishing some end especially by a computer.

- The word originates from “Algorism”:
  - a mathematical counting technique
  - Stemmed from the name of “Muhammad ibn Musa al-Khwarizmi”, an influencer mathematician

- “Al-khorezmi his background, his personality, his work and his influence” by Heinz Zemanek, Springer LNCS, 1979, Algorithms in Modern Mathematics and Computer Science, pp 1-81
  (https://link.springer.com/chapter/10.1007/3-540-11157-3_25)
Example Algorithm

step 1

step 2

=  

+  

step 3

step 4

step 5  

step 6
How algorithms rule the world

The NSA revelations highlight the role sophisticated algorithms play in sifting through masses of data. But more surprising is their widespread use in our everyday lives. So should we be more wary of their power?
How algorithms rule the world

The NSA revelations highlight the role sophisticated algorithms play in sifting through masses of data. But more surprising is their widespread use in our everyday lives. So should we be more wary of their power?

New algorithms may revolutionize drug discoveries—and our understanding of life

February 6, 2017
An economy of algorithms

ComputeFest 2017 tackles the future of the computational economy

By Leah Burrows | January 27, 2017
Robo-advice using algorithms are replacing financial planners
Big Data Algorithms Optimize Oil Wells

Ambyint has added a new product to a lineup geared toward taking the analysis out of data analytics, reducing labor costs associated with operating oil wells.
Algorithms Probably Caused a Flash Crash of the British Pound

Trading software may have overreacted to tweets about the French president's comments on Brexit.

by Jamie Condliffe  October 7, 2016

Overnight, the British pound dropped by 6 percent, to $1.13. Analysts are pointing the finger at an increasingly familiar financial scapegoat: the algorithm.
Scope of Computational Algorithms

- Computability
- Complexity
Scope of Computational Algorithms

Computability

Decide on problem computability:

- What problems can be solved by a computer?
- Can a computer solve any problem, given enough time and storage space?

Complexity
A computationally infeasible problem

input n
assume n>1
while (n != 1) {
    if (n is even) n = n/2
    else n = 3*n+1
}

A computationally infeasible problem

input n
assume n>1
while (n != 1) {
    if (n is even) n = n/2
    else n = 3*n+1
}

Is this problem terminates for all possible n>1?
A computationally infeasible problem

input n
assume n>1
while (n != 1) {
   if (n is even) n = n/2
   else n = 3*n+1
}

› Is this problem terminates for all possible n>1?
› We cannot write a computational algorithm to answer this question
Scope of Computational Algorithms

Decide on problem computability:
- What problems can be solved by a computer?
- Can a computer solve any problem, given enough time and storage space?

Analyze a computational algorithm performance:
- How fast can we solve a problem using a computer?
- How little storage space can we use to solve a problem?
- Design better algorithms.
Scope of Computational Algorithms

Decide on problem computability:
- What problems can be solved by a computer?
- Can a computer solve any problem, given enough time and storage space?

Analyze a computational algorithm performance:
- How fast can we solve a problem using a computer?
- How little storage space can we use to solve a problem?
- Design better algorithms.
Correct Algorithm

- A correct algorithm has two conditions:
  - Halts/terminates
  - Produces a correct output set for all possible input sets

- Will detail later on analyzing correctness of algorithms.
Grading and Policies

Course work
- Class participation (3%)
- Homework assignments (30%)
- Two mid-term exams (33%)
- Final inclusive exam (34%)

Delivery policies:
- The default late policy: submission allowed for 20% penalty for a calendar day.
- Assignments should be computer-typed.

Cheating is not allowed and will be reported
- If you are using any external source, you must cite it and clarify what exactly got out of it.
- You are expected to understand any source you use and solve problems in your own.
Reference Book

Course Content

- Introduction to Computational Algorithms
- Analysis of Algorithms
  - Divide and Conquer
  - Greedy Algorithms
  - Dynamic Programming
- Advanced Data Structures: Graphs
- Introduction to Advanced Topics: NP-Completeness
Credits

- Prof. Guy Blelloch notes

- Prof. Donald Knuth book
  - The Art of Computer Programming, Volume 1

- Prof. Madhusudan Parthasarathy notes
  - https://courses.engr.illinois.edu/cs373/sp2010/lectures/slides-lec1.pdf

- BBC Ideas
  - https://www.youtube.com/watch?v=oRkNaF0QvnI