Lecture 00

Course Overview

DATA STRUCTURES AND ALGORITHMS

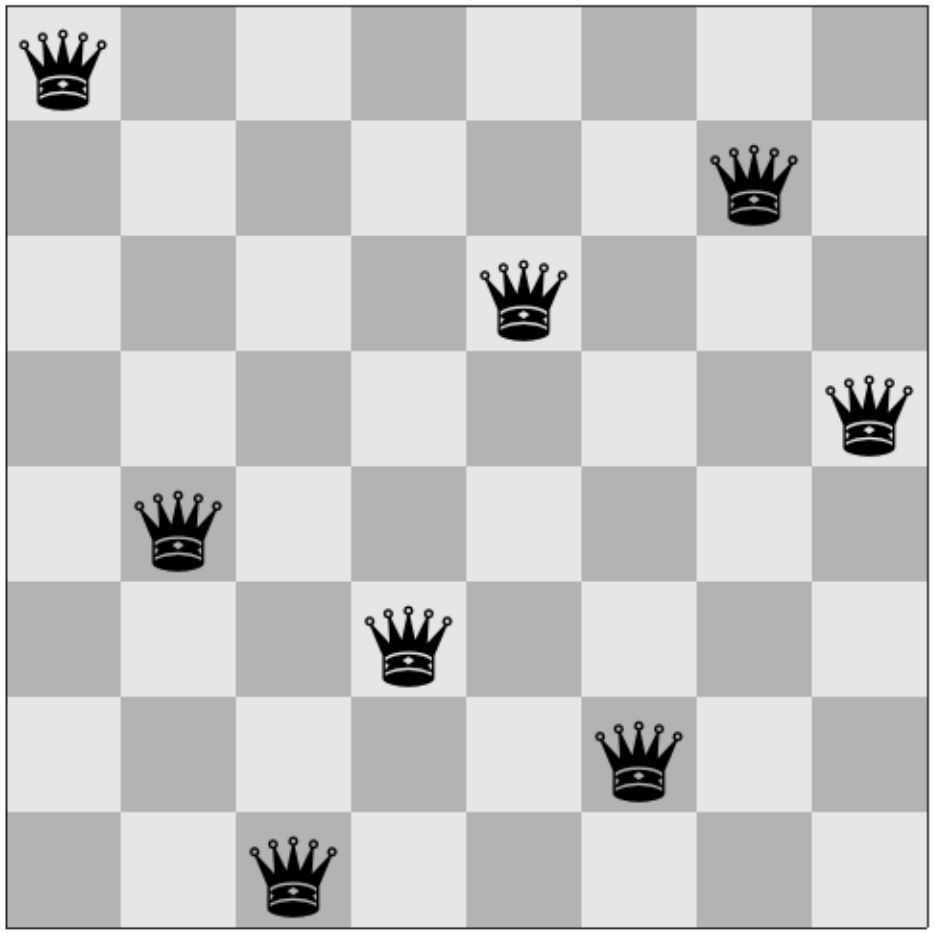
Fall 2017

University of Windsor

By Dr. Sherif Saad

N-Queen Problem

Given an NxN chessboard, arrange n queens so that none is attacking another.



N-Queen Problem

- What are the data that we need to store?
- What is the most efficient way to store this data?
- How could we solve this problem?
- What are the steps to solve this problem?
- Could we come up with a general solution?
- Is this general solution useful to solve other problems?

Job Interview: Level 1

1. What are linear and nonlinear data structures, give examples?
2. Which data structures are used for BFS and DFS algorithms?
3. Which Data Structure Should be used for implementing LRU cache?
4. Describe binary tree and its property?
5. Which data structure is used for a dictionary and spell checker?
6. What are the various operations that can be performed on different Data Structures?
7. How is an Array different from Linked List?

Job Interview: Level 2

- Given a file full of these strings (Netflix):
  - Vivek 1
  - Vivek 2
  - Ajit 3
  - Krishna 4
  - Keshav 6
  - Keshav 7

Print the duplicates and the IDs:

  - Vivek: 1, 2
  - Keshav: 6, 7

Job Interview: Level 2

1. What is a priority queue? How will you implement it? What is the complexity of each implementation? (Amazon)
2. Given a dictionary of words. There can be duplicates. Given a bag of letters. There can be duplicates. Print the length of the longest valid dictionary word that can be formed from the bag. (Amazon)
3. Describe the data structure that is used to manage memory (Google)
4. Given a binary tree, programmatically you need to prove it is a binary search tree. (Google)

Job Interview: Level 3

- How to implement an auto complete, like if I type, bo, it would suggest me boy, bow … etc. (Microsoft)
- How would you extend the algorithm to support word based auto suggestion i.e. if I search 'I am' it would suggest 'I am a boy' , ‘ I am good' etc. (Microsoft)
- Write a function that takes a pointer to the head of a linked list and an index and deletes that node. Node consists of a void pointer data and link to the next node. Write test cases for this function. (Microsoft)

Course Information

- CS60-254 Data Structures and Algorithms
- Prerequisite: CS60-100 and CS60-141
- Commitment:
- 3 hours lecture a week,
  - 1.5 laboratory hours a week,
  - 5-7 hours self study per week
- Course Web Page & Email:
  - Course Email: cs254@wasplabs.ca
  - Web Page: https://github.com/ebinsaad/CS60_254

Course Information

- **Lectures**
  - Tuesday and Thursday (2:30-3:50 PM) - Room Erie Hall 3123
- **Labs**
  - Monday (11:30 – 12:50 PM) - West Library Room 305 – Sec 51
  - Monday (2:30 – 3:50 PM) - Erie Hall Room 3119 – Sec 53

  - Wednesday (11:30 – 12:50 PM) - West Library Room 305 – Sec 52
  - Wednesday (2:30 – 3:50 PM) - Erie Hall Room 3119 – Sec 54

Course Topics

- Linear Data Structure
  - Arrays & Linked Lists
  - Stacks
  - Queues
- Non-Linear Data Structure
  - Trees
  - Graph
- Applying Data Structure:
  - Sorting and Searching Techniques

Course Topics

- Algorithm Design Paradigms
  - Backtracking
  - Divide and Conquer
  - Dynamic Programming
  - Greedy Algorithms
- Algorithm Analysis
  - Types of Analysis
  - Complexity Classes

Course Objectives

1. Teach fundamental data structures models and concepts.
2. Introduce the students to basic algorithm design and analysis techniques
3. Teach techniques to select appropriate data structure and algorithm design method for a specified application
4. Teach different algorithm design paradigms such as divide and conquer, greedy algorithms, dynamic programming.
5. Explain how to apply linear and nonlinear data structures to solve different problems
6. Expose the students to advanced data sorting and searching techniques.

Learning Outcomes

1. Assess the pros and cons of using a specific data structure to solve a particular problem.
2. Implement basic operations like searching, insertion, deletion, traversing on various data structures.
3. Conduct basic algorithm time and space complexity analysis.
4. Apply different algorithm design paradigms such as (divide and conquer, greedy algorithms, dynamic programming, etc.) to implement effective solutions for complex problems.

Course Schedule

Week-01	September 5, 2017	September 7, 2017	Course Overview
	Introduction To DS and Algorithms		
Week-02	September 12, 2017	September 14, 2017	Linear Data Structure
	Arrays and Linked Lists		
Week-03	September 19, 2017	September 21, 2017	
	Stacks		
Week-04	September 26, 2017	September 28, 2017	
	Queues		
Week-05	October 3, 2017	October 5, 2017	Non-linear Data Structure
	Trees		
Week-06	October 10, 2017	October 12, 2017	
	Trees & Graph		
Week-07	October 17, 2017	October 19, 2017	
	Graph	Midterm Exam	
Week-08	October 24, 2017	October 26, 2017	Applying Data Structures
	Searching and Sorting		
Week-09	October 31, 2017	November 2, 2017	Algorithm Design Paradigms
	Algorithm Desing Techniques		
Week-10	November 7, 2017	November 9, 2017	
	Divide and Conquer		
Week-11	November 14, 2017	November 16, 2017	
	Greedy Algorithm	Back Tracking	
Week-12	November 21, 2017	November 23, 2017	
	Dynamic Programming		
Week-13	November 28, 2017	November 30, 2017	Algorithm Analysis
	Complexity Classes		
Week-14	December 5, 2017	December 7, 2017	Course Conclusion
	Review		

Evaluation and Important Dates

- 30% There are 5 Lab Assignments (6% each)
- 25% Midterm on **October 24, 2017**
- 10% Practical Lab Exam on **November 27 and 29, 2017**
- 35% Final Exam on **December 14, 2017**

Evaluation and Important Dates

- Labs
  - Labs will run on Mon and Wed every week, and will start on Sep 18, and will finish on Nov 29.
  - There will be 11 lab sessions. In each odd-numbered lab session (1, 3, 5, 7 and 9), the lab assignment will be explained, and students will start working on it.
  - Lab assignments will be submitted in even-numbered lab sessions (2, 4, 6, 8 and 10).
  - The 11th lab session will be a one hour practical lab exam that is worth 10% of the course grade.
  - Each lab assignment is worth 6% of the course grade. If you finish earlier, namely in the corresponding odd-numbered session, you can submit the lab at that time and do not have to come to the next session (the corresponding even-numbered lab).

Course Instructor & TAs

- Course Instructor: Sherif Saad
  - PhD in Computer Engineering, University of Victoria 2015
  - MSc in Computer Science, 2007
  - BSc in Computer Science, 2003
- LinkedIn: https://www.linkedin.com/in/ebinsaad/
- Course Email: cs254@wasplabs.ca
- Office Hours @ Lambton Tower Room 5106
  - Thursday 4:15 – 5:15 PM ,
  - Monday 10:30 –11:30 AM or by appointment

Course Instructor & TAs

- Course TAs and Lab Instructors
  - Adam Barron barronh@uwindsor.ca
  - Jose Alba albaj@ywindsor.ca
  - Patrick Malolepszy malolepp@uwindsor.ca
  - Bonaventure Molokwu molokwub@uwindsor.ca
  - Jack Vanier vanier1@uwindsor.ca

Next Class

- Introduction To DS and Algorithms
- Arrays

Questions??