VE281
Data Structures and Algorithms

Introduction

What will You Learn

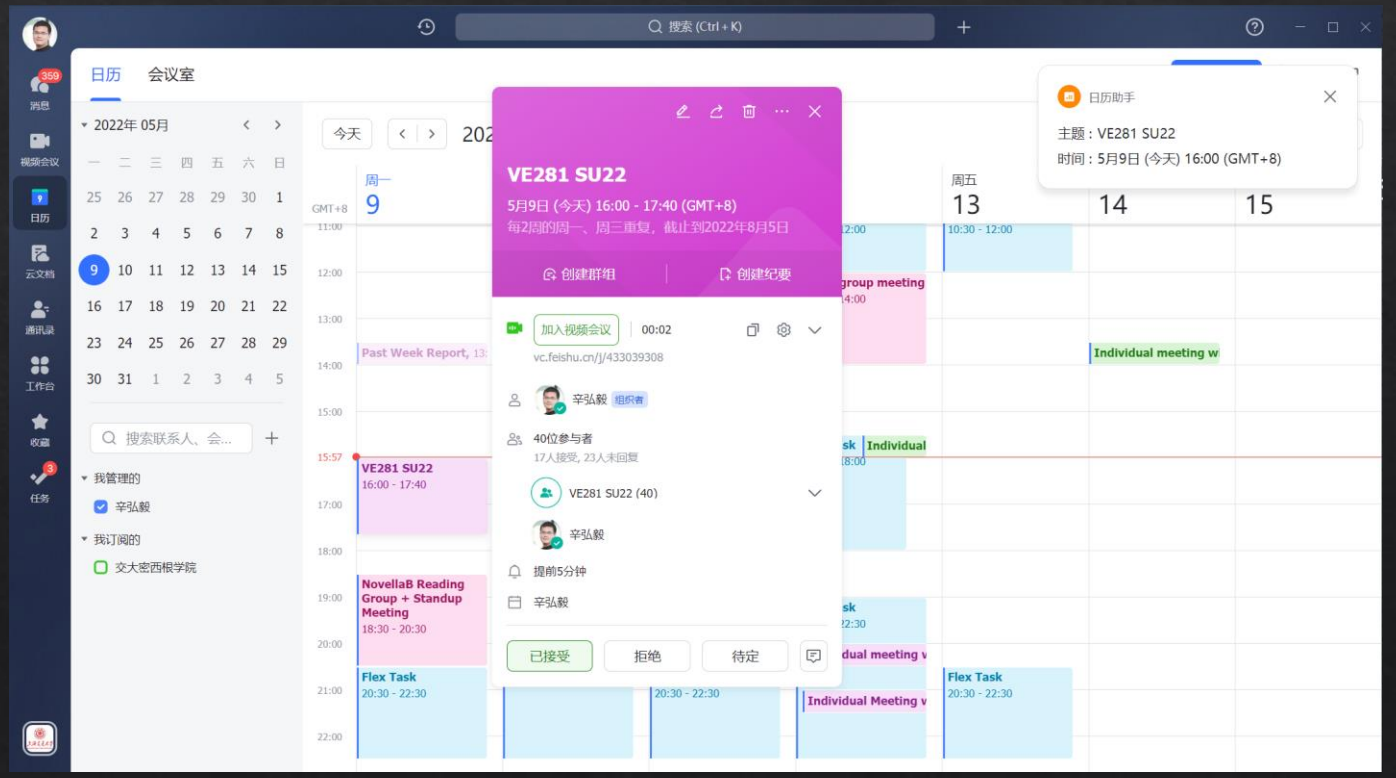
- Algorithms
  - The idea
  - Their efficiencies
- Discrete math
  - How to evaluate their efficiencies
- Hands on experience
  - How are they implemented
  - Real world applications

Outline

- Course logistics
- Introduction

Time and Location

- Time: Monday 4:00-5:40 pm, Wednesday 4:00-5:40 pm, and Friday 4:00-5:40 pm (odd weeks)
  - Arranged on Feishu
- Location: D-107 (Mon), D-412 (Wed/Fri)



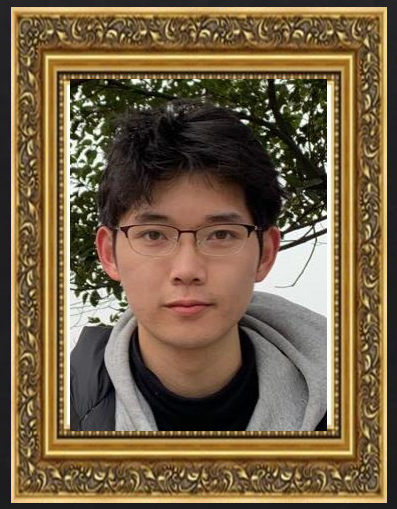
Instructor

- Hongyi Xin
  - 2007 JI alumni
  - CMU CS PhD
- Email: hongyi.xin@sjtu.edu.cn
- Office hour
  - Flexible, please contact me on Feishu

Teaching Assistants

- Yicheng Hou
  - Email: wonder0702@sjtu.edu.cn
- Zining Wang
  - Email: wznmickey@sjtu.edu.cn
- Siyuan Lin
  - Email: jeremy.l@sjtu.edu.cn
- Xinyi Chen
  - Email: chenxinyi@sjtu.edu.cn






Textbooks for Reference (Not Required)

- "Data Structures and Algorithm Analysis," by Clifford Shaffer.
  Online available: http://people.cs.vt.edu/~shaffer/Book/C++3e20120605.pdf
- "Algorithms," by S. Dasgupta, C. Papadimitriou, and U. Vazirani.
- "Introduction to Algorithms, 3rd edition," by Thomas Cormen et al., MIT Press, 2009.
- "Data Structures and Algorithms with Object-Oriented Design Patterns in C++" by Bruno Preiss.

Grading

- Composition
  - Participation: 5%;
    - It is your job to get me know about you
    - Discuss on piazza; ask questions during breaks; ask and answer questions in class
  - 4 written assignments: 20%
  - 4 programming assignments: 30%
  - 1 final programming assignment up to 2% bonus
  - Midterm exam (written): 20%
  - Final exam (written): 25%
- We will curve the final grades
- Questions about the grading?
  - Must be mentioned to the instructor or the TAs within one week after receiving the item

Programming Assignments

- Your code must be compatible with GNU-g++
  - Not the Apple LLVM g++!
- C++11, C++14 and C++17 standards are allowed
  - Compile with the option `--std=c++11, --std=c++14, --std=c++17`
- Do not copy code from github
- Do not post code on github
  - I take honor code very seriously
- Turn in through the online autograder
  - https://joj.sjtu.edu.cn

Assignment Deadline

- Each written assignment must be turned in on Canvas in PDF format
- Each Programming Assignment (PA) must be turned in by 11:59 pm on the due date to be accepted for full credit.
  - However, we still allow you to submit your PA within 3 days after the due date, but there is a late penalty.

Hours Late	Scaling Factor
(0, 24]	80 %
(24, 48]	60 %
(48, 72]	40 %

  - No PA will be accepted if it is more than 3 days late!

Assignment Deadline

- In **occasional** cases, we accept deadline extension request.
  - Contact **ME**, not TAs!
  - Tell me early! The earlier you let me know the more likely I will accommodate to your case!
  - **ONLY** be granted for **documented** medical/personal emergencies or **Academic** reasons
  - **NOT** granted for reasons such as accidental erasure/loss of files and outside conflicting commitments
  - If you experience any issues with the online autograder, please contact me or TAs

Some Suggestions

- Attend the class
  - Ask and answer questions
- Start doing the homework early!
  - Don’t wait until the last minute.
- Back up your code frequently in case your accidentally deletes your code files.
  - In real world, if you accidentally lose your code, your supervisor would not care for excuses! Have good habits!

Get the Most out of a Lecture

- Information breakdown:
  - Verbal communication is a linear process.
  - You cannot understand the rest of the lecture if your missed a key concept.
  - Ask questions!
- Lecture format:
  - Ask questions during short breaks (ask out loud and in turn).

Exams

- Written exams.
  - Some short questions
  - Some algorithm design problems
  - The question will mimic real world algorithm problems
- Closed book and closed notes

- No electronic devices are allowed
  - These include laptops and cell phones
  - You can bring calculators but you shouldn't need them

- If we go online, here are the setups:
  - You are required to setup 2 cameras while doing the exam
  - Write your answer on an A4 paper sheet
  - Take pictures and submit them through canvas

Collaboration and Cheating

- You can discuss the homework with your classmates but not sharing answers
- You must finish all the assignments yourself
- Some behaviors that are considered as cheating:
  - Reading another student's answer/code, including keeping a copy of another student's answer/code
  - Copying another student's answer/code, in whole or in part
  - Having someone else write part of your assignment
  - Using test cases of another student
  - Testing your code with another one's account (Testing chances are limited)
  - Keep your code safe! (You are also responsible if your code is leaked)

"**Another student**" includes a student in the current semester or in the **previous** semester.

Collaboration and Cheating

- The previous lists of behaviors are **deliberate** cheating, but some **unintentional** actions could make you look like cheating. For example,
  - Using other people's code to test the autograder
- You should be extremely careful!
- Do not share photos of your code!
- Do not post your code to github!
- Do not copy code from github!

Collaboration and Cheating

- You should be responsible for all answers/codes you submit.
- If you submit a copy of another student's work (or overwrite another student's work), your case will be submitted to the Honor Console

- Any suspect of cheating will be reported to **the Honor Council at JI**.
- For programming assignments, we will run an automated test to check for unusually similar programs. Those that are highly similar - in whole or in part - will be reported to **the Honor Council at JI**.
- **Penalty** of honor code violation

1. Reduction of the grade for this assignment to 0, **plus**
2. Reduction of the final grade for the course by one grade point, e.g., B+ → C+, for **both students** involved

Getting Help

- If you have any technical questions, come to see TAs and instructor during the office hour!
  - Answering technical questions through email is inefficient and I will only answer them during office hours.
  - Post questions on Piazza.
  - Answer other student's questions (counts toward participation)

Canvas

- Log into Canvas: https://umjicanvas.com

- Check the class webpage on the Canvas regularly for
  - Announcements
  - Slides
  - Assignments

- Course slides will be uploaded onto Canvas before each lecture

Prerequisite

- Ve280 Programming and Elementary Data Structures
  - Compiling and debugging on Linux operating systems
  - C++ programming, including pointers, arrays, structs, etc.
  - Recursion
  - I/O streams, including file I/O
  - Classes
  - Virtual functions
  - Dynamical memory management
  - Template
  - How to implement a linked list, stack, queue

Prerequisite

- Ve203 Discrete Mathematics
  - Computational complexity analysis
  - Some basic sorting algorithm, e.g., bubble sort, insertion sort, merge sort
  - Divide-and-conquer algorithm, master theorem
  - Graph, graph representation, depth first search, Dijkstra's algorithm (shortest path)

- Some important concepts will be reviewed

References and Copyright

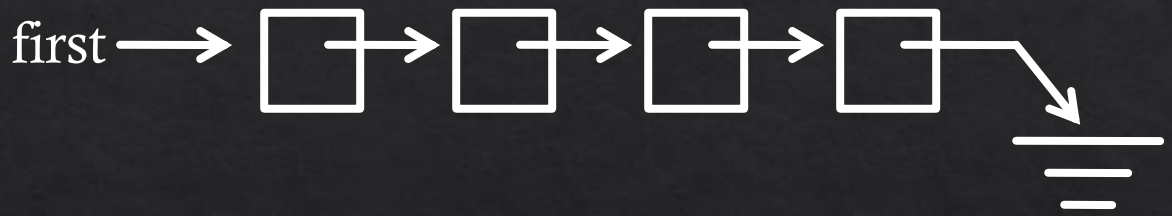
- Slides used (modified when necessary)
  - Weikang Qian, JI, SJTU
  - Sugih Jamin, University of Michigan
  - Sartaj Sahni, University of Florida
  - Bert Huang, Columbia University
  - Tim Roughgarden, Stanford University
  - Clifford Shaffer, Virginia Tech

Outline

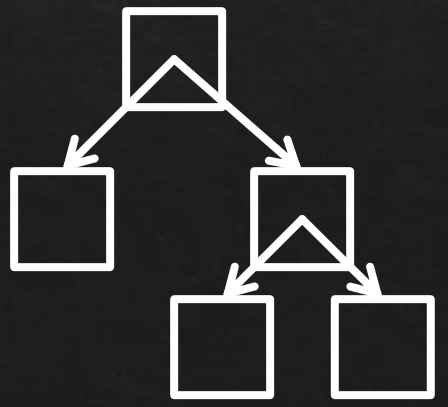
- Course logistics
- Introduction

Data Structures and Algorithms

- Data structure is a particular way of organizing data in a computer so that it can be used efficiently.
  - Example: linked list



- We can store a set of records as a linked list
  - or as a tree (to be talked later).

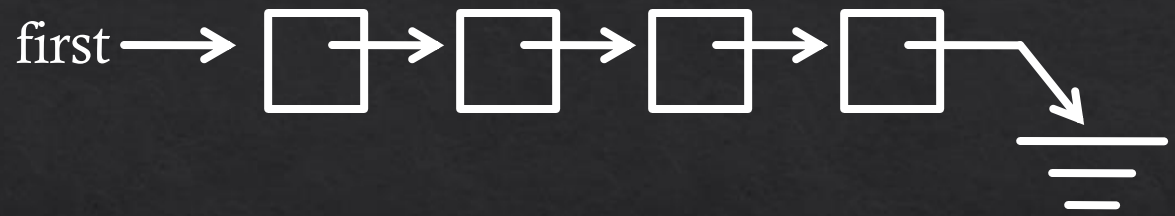


Logical versus Physical Form

- A data structure have both a **logical** and a **physical** form.
- Logical form: definition of the data structure at an abstraction level.
- Physical form: implementation of the data structure.

Data Structure Example: Linked List

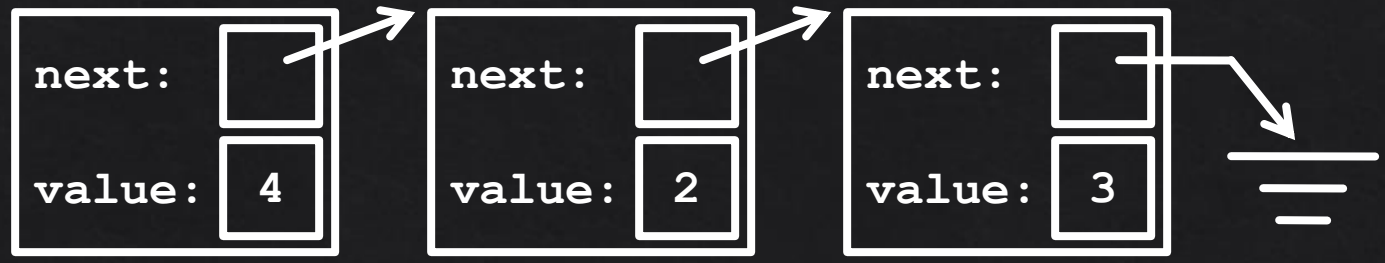
Logical Form



Physical Form

```
class IntList {
  node *first;
public:
  ...
};
```

```
struct node {
  node *next;
  int   value;
};
```

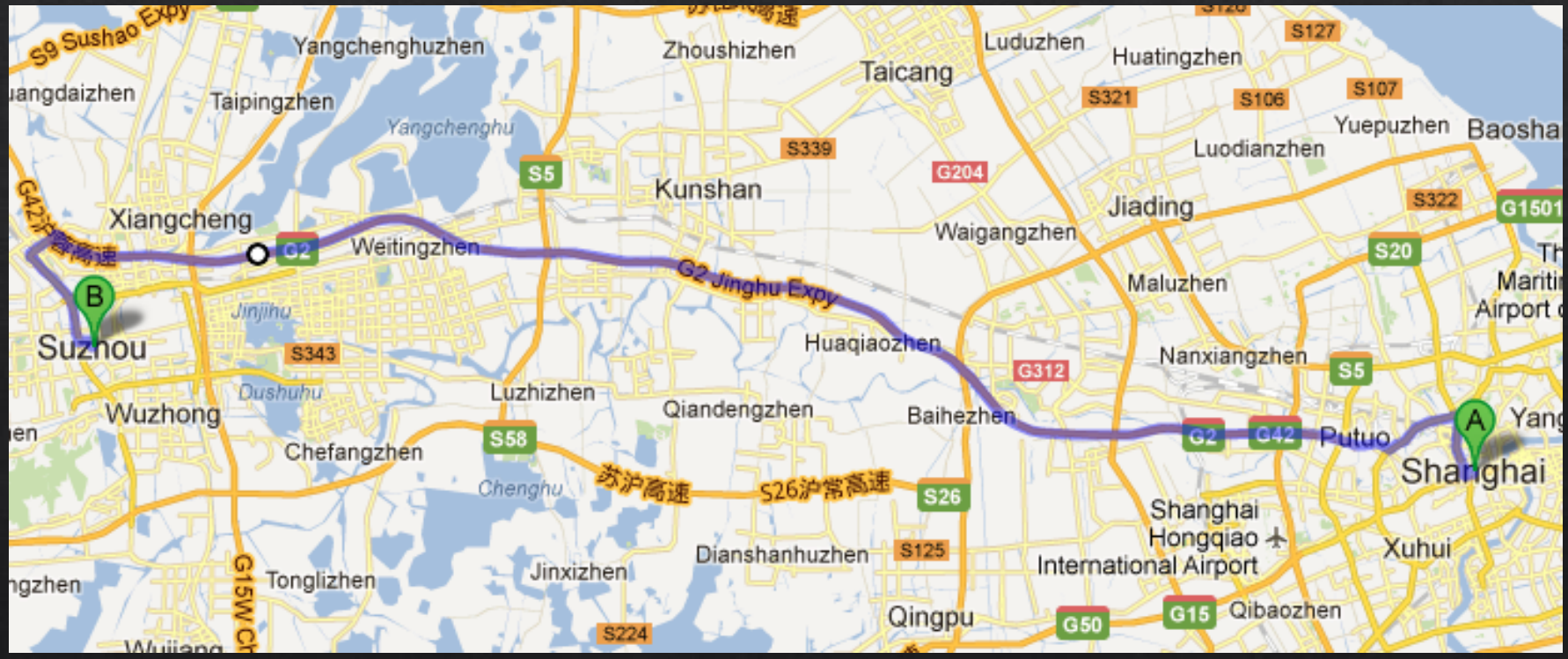


Data Structures and Algorithms

- Data manipulation requires an algorithm – a sequence of steps that solve a specific task
- Data structures + Algorithms = Programs
- The study of data structures and algorithms is fundamental to Computer Science.
  - Database related to balanced binary search tree.
  - Computer networks related to shortest path algorithm.
  - …

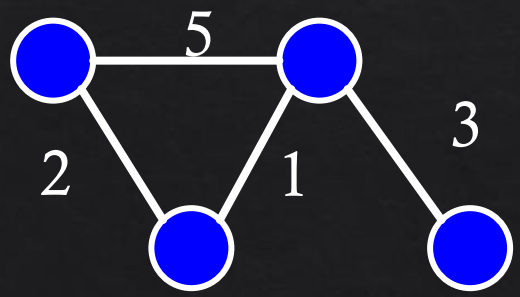
Real World Problem: Navigation

- Finding the shortest route from Shanghai to Suzhou



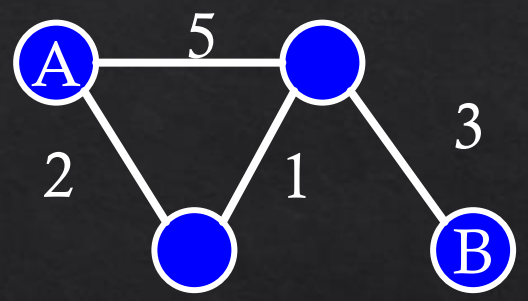
Real World Problem: Navigation

- What information do we need?
  - Streets.
  - Intersections of streets. (We assume that our departure place and destination are at certain intersections.)
- How do we store the information in computer?
  - Graph: consisting of “nodes” and “edges”.
  - Each edge has a weight to denote the distance between two nodes.
  - DS:
    - Adjacency list
    - Adjacency matrix



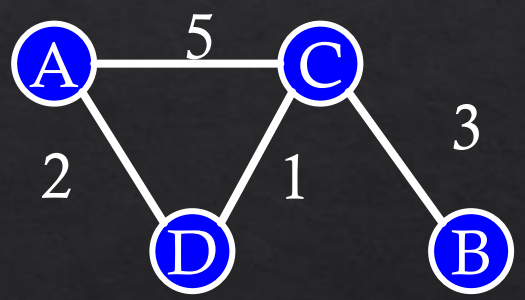
Real World Problem: Navigation

- The algorithm: finding the shortest path from a source node (A) to a sink node (B)
- Algorithms adapt to data structures



Challenges: Efficiency

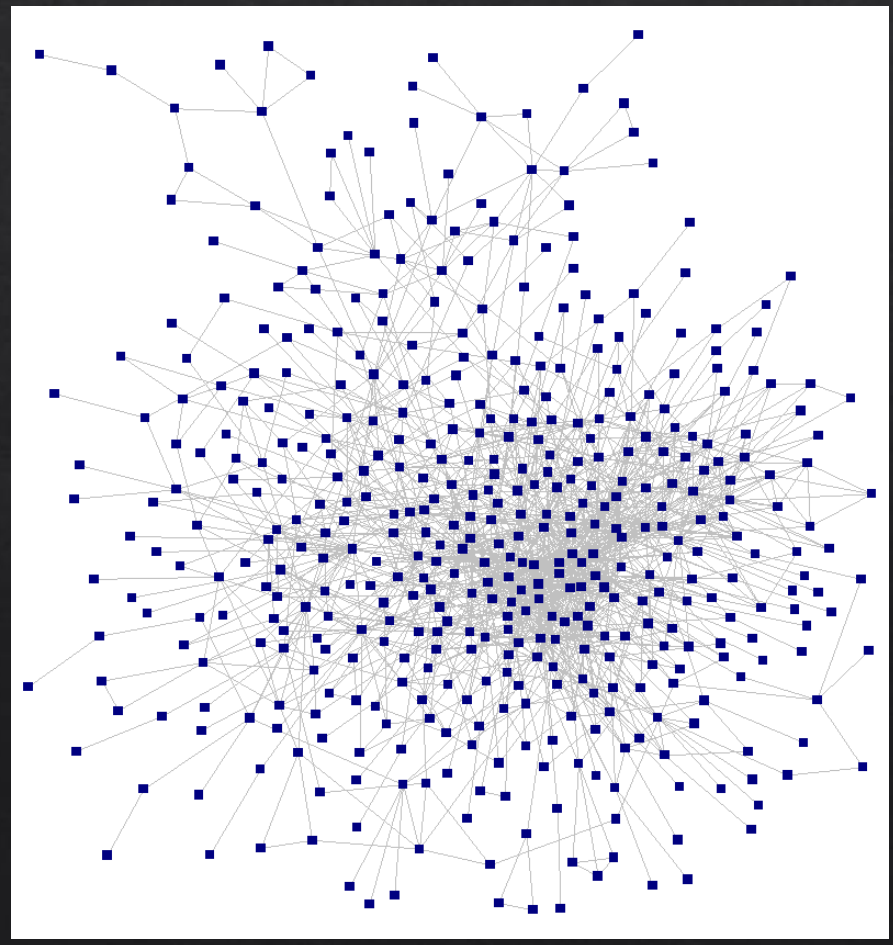
- For a small number of nodes, we can enumerate all the possible paths.



- Path A → C → B: 8;
- Path A → D → C → B: 6;
- The minimum is 6.

Challenges: Efficiency

- However, in real world, the graph is much more complicated.
- It is impossible to enumerate all the possible paths!
- How can we solve the problem?
  - Dijkstra's algorithm



More about Efficiency

- Choice of data structures or algorithms can make the difference between a program running in a few seconds or many days.
- Example: Number of comparisons for **linear search** and **binary search** (Worst Case)

Input Size	Linear	Binary	Ratio (L/B)
64	64	6	10.7
128	128	7	18.3
256	256	8	32
512	512	9	56.9
1024	1024	10	102.4

More about Efficiency

- A solution is said to be efficient if it solves the problem within its resource constraints
  - Space, i.e. memory consumption
  - Time

- The cost of a solution is the amount of resources that the solution consumes

✓ **Our major concern**

- We value efficiency of the data structures and algorithms!
- We will learn how to analyze their efficiency

Course Objectives

- Learn the tool:
  - Common data structures and algorithms
  - And their efficiency

- Apply the tool
  - Solve a problem using existing data structures and algorithms
  - Choose the right tool:
    - some tools are better for certain tasks than other tools
    - Do performance analysis

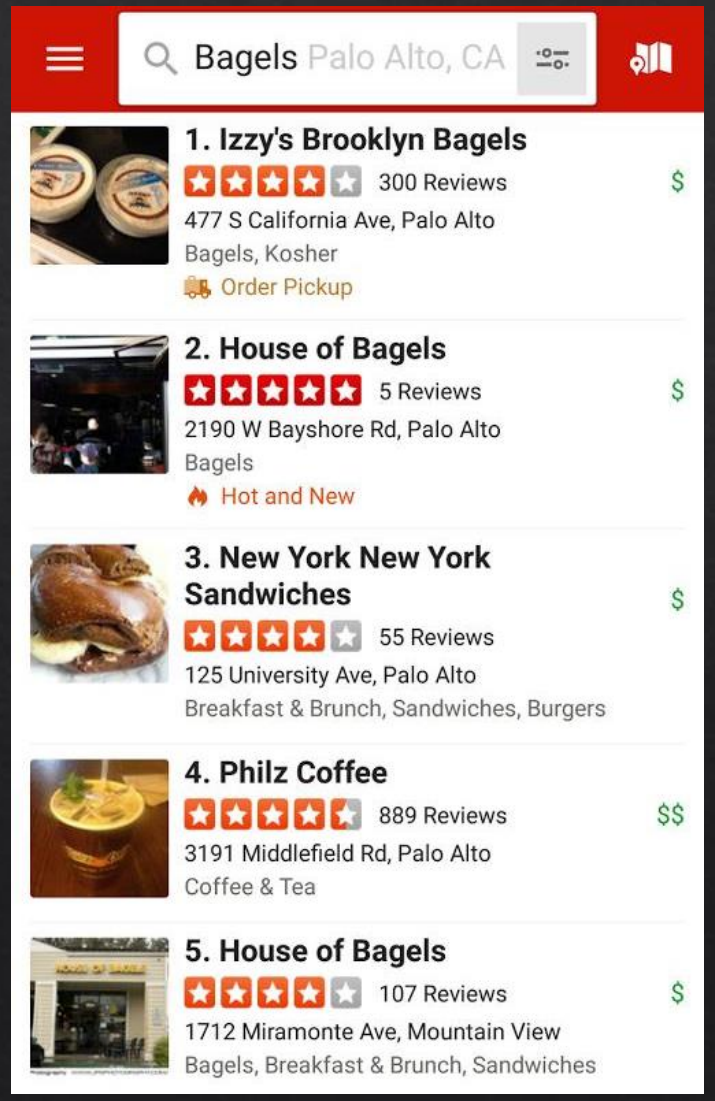
Topics

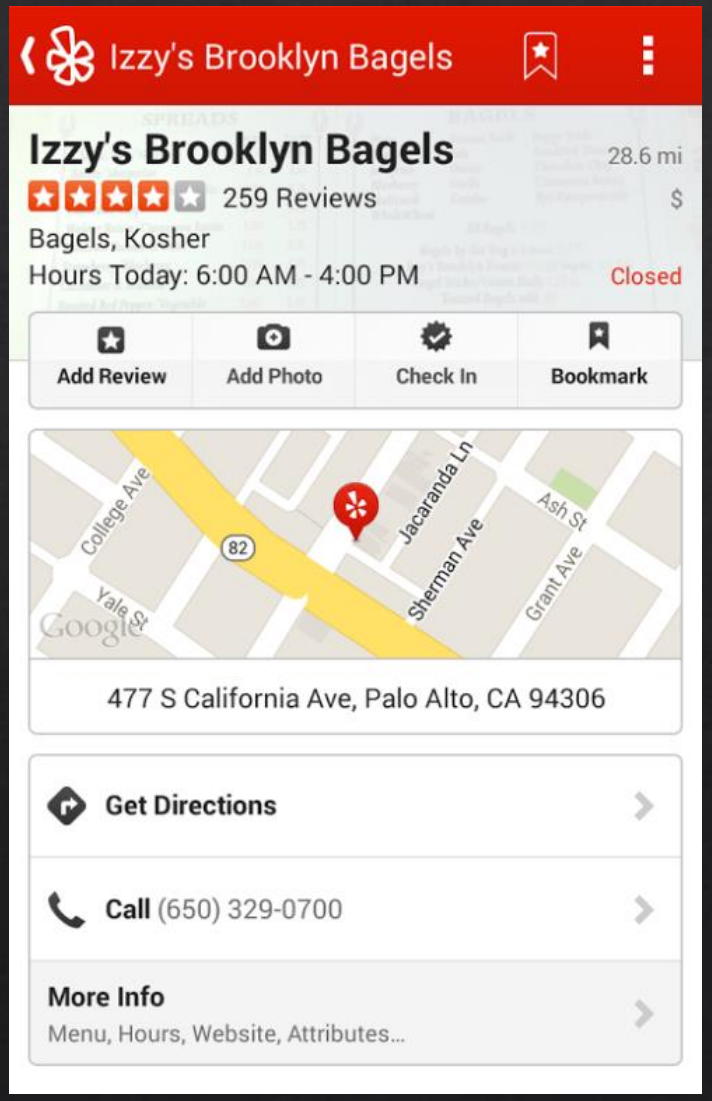
- Asymptotic Algorithm Analysis
- Data structures
  - Trees, including binary search tree, balanced binary search tree
  - Hash table
  - Heaps
  - Graphs
- Algorithms
  - Sorting and searching
  - Graph-related algorithms
    - minimum spanning tree
    - topological sorting
    - Shortest Path
  - Dynamic programming

Project: A Yelp-like Android app

- Sort
- Search
- K-D Tree
- Path Finding

- Not as pretty though!





Questions?