COSC 2306 Data Programming
Course Overview

Feng Yan

Computer Science

University of Houston

IDS Intelligent Data and Systems Lab

Feng Yan
Associate Professor of CS & ECE
PGH 589, fyan5@central.uh.edu

Research methodology

System Building
Algorithm Design
Characterization

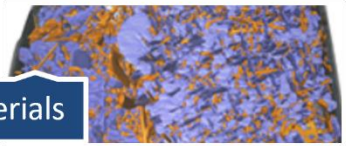
Automated, high-performing, efficient, robust, and user-centric *methodologies* and *systems* at scale

Big Data
Machine Learning
Systems

MLaaS	•*Design [NeurIPS'21, AAAI'21, KDD'20, AAAI'20-oral, BMVC'19, Middleware* •*Training [SOSP,21, ICSEW'21, IPDPS'20, ICDCS'18, NIPS'17-oral, KDD'15]* •*Serving [SC'20, TCC'20, IoT'20, ATC'19, SC'19, EuroSys'19, TNSM'18, SC'16]*	Workflows, new learning paradigms (GNN,CL,SR,R), serverless, explainable…
Federated Learning	•*Heterogeneity-aware FL system [HPDC'20]* •*Robustness to adversarial attacks [AAAI'21, SoCC'21]* •*Homomorphic Encryption (compression + batching) [ATC'20]*	Fairness, dropout, forgetting, holistic, NAS, HPO, robust, heterogeneity, vertical…
Cloud, HPC, Edge, IoT	•*Burstable computing [TNSM'21, Middleware'19, ICAC'18, CLOUD'17]* •*Scientific apps [IPDPS'22, SC'21, AAAI'20, TBD'20, CLOUD'18 (best paper)]* •*Scalability & efficiency [SC'21, VLDB'21, FAST'20, CLOUD'19 (best paper)]*	Serverless, Cross eco-sys (HPC+cloud+edge), sprinting, imbalance compression …

Interdisciplinary


Wildfire


Materials


Pavement

…

IDS Intelligent Data and Systems Lab

Feng Yan
Associate Professor of CS & ECE
PGH 589, fyan5@central.uh.edu

TA

Mina Yazdani

Email: myazdani@cougarnet.uh.edu

or minayazdani5@gmail.com

Office Hours:

- Monday 6pm-8pm, via Teams (no appointment needed)
- If you prefer different hours or an in-person meeting, just make an appointment

Learning Center

- **ConocoPhillips Computer Science Learning Center**
- **Where:** PGH 233 or MS Teams
- **When:** Mondays thru Thursdays
- **What:** tutoring (set up development environment, topic of each week)
- **How:** make an appointment via https://outlook.office365.com/owa/calendar/UHCSTutoringandLearningCenter1@UofH.UH.EDU/bookings/, walk-in is also possible but depends on availability

Office Hour

Better to talk to TA or Learning Center first, especially on assignments, but I also host office hours

- Wednesday 4pm-5pm (better to let me know in advance)
- Talk to me directly after each class

Email/Teams: fyan5@central.uh.edu

Meet with Each other

Let's meet with each other!

1. Your name?
2. Which program, which year?
3. Professional interests?
4. Others (personality, hobbies, stories…)

Prerequisites

- COSC 1336 Computer Science and Programming
- Basic programming skills
  - Designing, coding, debugging, documenting, executing programs
  - Use existing libraries, modules
  - Write structured programs
- Computational problem solving skills
  - Math
  - Reasoning
  - Basic knowledge of software and hardware

Course Scope

- Refresh of fundamentals of programming
- Python built-in data types and structures (e.g., lists, dictionaries)
- Advanced data structures: such as stacks, queues, linked lists, hash tables, trees
- Recursive algorithms
- Sort and search algorithms
- Debugging, testing, evaluating program speed

Tentative Course Topics

- Python Basics
- Functions, Lists, Dictionaries, Files
- OOP and Class
- Complexity
- Stacks
- Queue
- Linked Lists
- Recursion
- Search
- Sort
- Trees

Course Materials

Slides and assignments (distributed via Canvas)

Required textbook: no required textbook

Highly recommended textbook:

- *Problem Solving with Algorithms and Data Structures Using Python by Bradley N. Miller and David L. Ranum, Franklin, Beedle & Associates, 2011.*

Course Grading

Aug 25 – Dec 6, total 15 weeks, (30-2=28) classes

Component	Percentage
Classwork (in class activities/quizzes) and homework assignments	70%
Exam(s)	30%
Total	100%

Grade Scale:

90 or higher is an A
86 – below 90 A-
82 – below 86 B+
78 – below 82 B
74 – below 78 B-
70 – below 74 C+
66 – below 70 C
62 – below 66 C-
58 – below 62 D+
54 – below 58 D
50 – below 54 D-
0 – below 50 F

Quizzes

1. How: Canvas by using access code and LockDown Browser (bring a laptop with camera for each class)
2. When: during the class
3. What: content in this class and previous classes
4. Format: TBD (True or False, Multiple Choices)
5. Note: no quiz make-up (fair to everyone), but alternative make-up may be offered case by case

Respondus LockDown Browser

Respondus LockDown Browser is used for all quizzes/exams:

- Download and learn more: https://uh.edu/fdis/technology/respondus-lockdown/lockdown-tutorial-student/
- Please bring a laptop with working camera (you will be responsible for failed camera/laptop)
- Please download Respondus LockDown Browser and compete the webcam setup before starting the quiz/exam

Missing Class

Email both lecturer and TA before class unless it is not feasible (justify the situation)

Failure to notify before class may result in losing the make up opportunity

For valid reasons with proper proof, make up is offered (busy with other class, sleep over is NOT considered as valid reasons)

Missing Class

Make up for missing classes (valid reason + proper proof)

- **Write self-study notes of the missing class topic** (minimum of 2 pages, font size <= 11, line space = 1.0)
- **Credit back amount depends on the quality of self-study notes**
- **Frequent absence is determined case by case, make up is not guaranteed (missing 15% or more (> 4 classes) is typically not accepted)**
- **Submit within 1 week of missing class**
- **AI assistance is NOT allowed – TA will check and significant penalty would be applied if found**

Assignments

Homework Assignments

- Help you understand and practice things we learned in the class
- Most of them are coding
- Some of them are answering questions
- Assign and submit via Canvas

Exams

1. How: open-book (only paper-based materials such as notes/slides/books/documents, but no access to any app, no internet/AI assists, no discussion/communication in any kind)

2. When: TBD, 2 exams (one in the middle, one towards the end)

3. What: everything we covered in class

4. Format: TBD (True/False, Multiple Choices, Small Coding)

In Class Programming with Bonus

1. Solve programming problems during the class

2. Volunteers: earn 0.5-1 bonus credit for small and 1-2 bonus credits for big problems (even if not fully correct, get bonus credits)

3. Maybe allow another student to try if one fails, and other students can help (students who helped a lot may share the bonus credits)

Late Assignment Submission

Lose 10% credits of the assignment for each day after the deadline

Syllabus

Syllabus

Available on Simple Syllabus

Set Proper Expectations -- not an easy course

- If you perform poorly in prerequisite, **expect** that you might be suffering in this course
- This is an advanced course, so **don't expect** that knowing the syntax would be enough, you will learn some algorithms and logic thinking is critical here (yes, lots of content is **theoretical**)
- Attendance is required, and **don't expect** good grades without good attendance
- The assignments will be very time consuming, so **don't expect** good grades without significant time and efforts commitment

Why this Course Matters? -- critical in job interview & daily job

- If you know all the syntax and good at writing code, it is good but far from sufficient
- Problem-solving skills is the key – in computer science, it is a combination of skills in coding, algorithm, design, optimization, expertise in certain area etc.
- Companies pay you high salary not because you can write code (many children can do it), but solve challenging problems
- To pass coding interview and land an intern/job offer, the perquisite is the first step, this course is the second step, there are more steps ahead
- If you could not master the content in this course, it is highly likely you could not pass an intern/job interview

Interview Q1: Search (simple)

Given an array of integers `nums` which is sorted in ascending order, and an integer `target`, write a function to search `target` in `nums`. If `target` exists, then return its index. Otherwise, return `-1`.

You must write an algorithm with `O(log n)` runtime complexity.

Example 1:

> **Input:** nums = [-1,0,3,5,9,12], target = 9
> **Output:** 4
> **Explanation:** 9 exists in nums and its index is 4

Example 2:

> **Input:** nums = [-1,0,3,5,9,12], target = 2
> **Output:** -1
> **Explanation:** 2 does not exist in nums so return -1

Constraints:

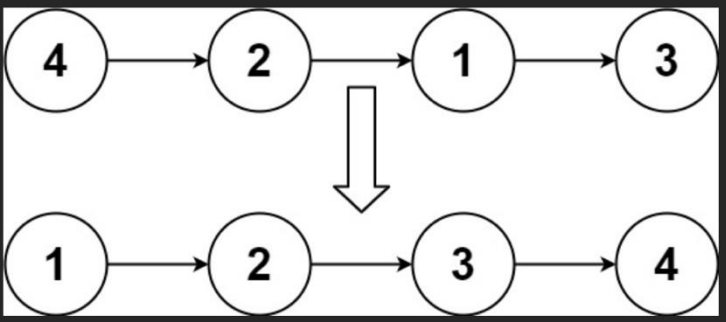
- `1 <= nums.length <= 10^4`
- `-10^4 < nums[i], target < 10^4`
- All the integers in `nums` are **unique**.
- `nums` is sorted in ascending order.

```python
def search(nums, target):
    left, right = 0, len(nums) - 1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            return mid
        elif nums[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

Interview Q2: Sort List (medium)

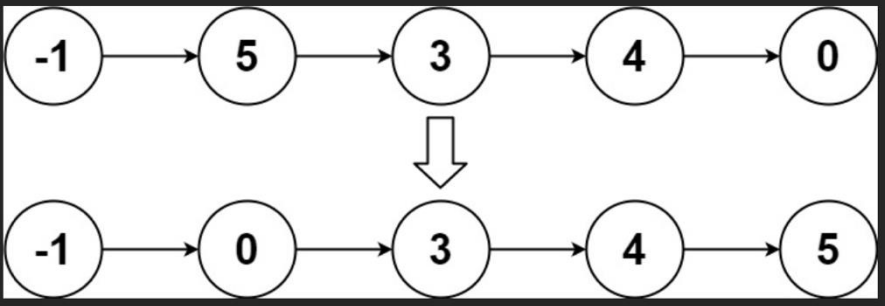
Given the `head` of a linked list, return *the list after sorting it in* ***ascending order***.

Example 1:




```
Input: head = [4,2,1,3]
Output: [1,2,3,4]
```

Example 2:




```
Input: head = [-1,5,3,4,0]
Output: [-1,0,3,4,5]
```

Example 3:

```
Input: head = []
Output: []
```

Constraints:

- The number of nodes in the list is in the range `[0, 5 * 10^4]`.
- `-10^5 <= Node.val <= 10^5`

Follow up: Can you sort the linked list in `O(n logn)` time and `O(1)` memory (i.e. constant space)?

THANK YOU!

Questions?