
Tech Interview Prep

Lecture 2

Christian Yongwhan Lim
Thursday, September 7, 2023

Christian Yongwhan Lim



Education



Part-time Jobs



Full-time Job



Workshops



Coach/Judge



<https://www.yongwhan.io>

Instructor's (Terse) Background

- Email: yongwhan.lim@columbia.edu
- **Associate in Computer Science** at Columbia;
- **ICPC Head Coach** for Columbia University;
- **Internship Manager** at ICPC Foundation;
- **ICPC North America Leadership** Team;
- **CEO** (Co-Founder) in a Stealth Mode Startup;
- **Owner** in Christian and Grace Consulting LLC;
- **Visiting Instructor** at Cornell-Tech;



<https://www.yongwhan.io>

Wonderful TAs!

- **Akash Nayar**

- Email: akn2120@columbia.edu
- Administrative tasks (e.g., auto-grading, etc)

- **Suro Lee**

- Email: sl5203@columbia.edu
- Office Hour: **Thursdays**, from **4pm ET** to 5pm ET @ Mudd 1st floor

Akash Nayar (TA)

- SEAS Senior
- 1.5 years of Competitive Programming experience
- Interested in Machine Learning and AI
- Python -> C++ convert (for CP)



Suro Lee (TA)

- 2nd Year MSCS Student
- Previously a SWE at Samsung
- Interested in ML, Competitive Programming, and anything else that will get me a full-time job



Wonderful Course Designer!

- **Grace Lim**
 - Email: gc3000@columbia.edu

Grace Lim (Course Designer)

- GS Junior
- Worked in Google before: at Google Play for ~4 years.
- Won a t-shirt from TopCoder Open before.
- I love CS and Psychology!
- Will bring a lot of enthusiasms here!



Point Clarifications

- You need **150** points by the end of the semester to receive a full mark on the assignment portion of the final grade (60% of the final grade).
- 1 AC live submission in LeetCode Weekly or Biweekly = **1 point**
 - This will start **Saturday, September 9, 2023.**
- 1 AC submission in TechPrep AI Daily Challenge = **1 points**
 - This will start **Saturday, September 16, 2023.**

(Auto-) Excused Absences

- Up to **2 lectures** are automatically excused.
 - The exception is a date you are presenting.
- Beyond that, it will lead to an automatic failure.

Course Contract

- Submission deadline is **THIS FRIDAY (TOMORROW)**!
- **Please complete it before you forget!**
- Any clarification questions on the contract? **Is policy on "proper citation" clear enough?**

Module II Presentation Spreadsheet

- Please **CHECK** your Module II presentation slot [here](#).

Module II Presentation Spreadsheet

- Please **CHECK** your Module II presentation slot [here](#).

Week #	Lecture #	Date	Student #1	Student #2	Student #3	Student #4
Week 4	Lecture 7	9/26/2023	Brendan (jbf2173)	Aparna (ak4605)	Caden (cl4319)	Nick (hl3648)
	Lecture 8	9/28/2023	Chenhao (cl4298)	Xin (xb2165)	Wo (wl2834)	Serdar (sm5038)
Week 5	Lecture 9	10/3/2023	Tsai-Chen (th2990)	Youngseo (yl4652)	Richa (rvb2121)	Shreyas (sak2286)
	Lecture 10	10/5/2023	Racquel (rl3168)	Jennifer (jd3794)	Divyang (dm3880)	Gyuseok (gp2693)
Week 6	Lecture 11	10/10/2023	Brennen (btb2125)	Allison (al4130)	Justin (jfp2130)	Katya (kpr2120)
	Lecture 12	10/12/2023	Charlie (cs4206)	Patrick (pt2556)	Jeannie (jr3766)	Joshua (jz3311)
Week 7	Lecture 13	10/17/2023	Gabriel (ggt2112)	Ethan (ec3665)	Jacklyn (ct3064)	Jesse (jv2761)
	Lecture 14	10/19/2023	Ming (mw3705)	Qianyi (qw2324)	Joey (zx2325)	Abhinav (ag4786)
Week 8	Lecture 15	10/24/2023	David (djl2197)	Patrick (psy2107)	Yurui (yc3699)	Elena (eyz2003)
	Lecture 16	10/26/2023	Yan (yc3855)	Yaqi (yz4341)		

Module II Presentation Spreadsheet

- The spreadsheet is auto-populated; I selected the slots through randomization.
- If you have an irreconcilable conflict or spot a typo (sorry, if I did!), please send me an email to change the date!

Ed Discussion Forum

- Please make a **proactive** use of Ed Discussion tab in Courseworks.
- Feel free to ask questions (and provide answers) freely :)

Office Hours

- By appointment through <https://calendly.com/yongwhan/quick-chat-blitz>.

Roll Call

- While roll call is in progress, please complete the Google form here: <https://bit.ly/tech-prep-survey-2>.
- A way to get to know you, but most likely just in the first week!
- Fill free to continue working on the Google form above while I answer your questions from lecture 1 survey!

Selected Questions from Lecture 1 Google Form (1/5)

- Will the questions solved in the class cover major interview questions?
 - Yes, absolutely!
 - We will cover:
 - **Lecture 1:** Primitive Types;
 - **Lecture 2 (Today!)**: Array; String; Linked List; Stack and Queue;
 - **Lecture 3:** Binary Tree; Heap; Sorting;
 - **Lecture 4:** Binary Search; Binary Search Tree; Hash Table;
 - **Lecture 5:** Recursion; Dynamic Programming;
 - **Lecture 6:** Greedy Algorithm and Invariant; Graph;

Selected Questions from Lecture 1 Google Form (2/5)

- **Can one audit this course?**
 - No, due to a space limitation.
- **Can the classes be recorded?**
 - No recording.
- **How many hours do you expect students to spend on this course per week? Is 30 hrs per week enough?**
 - Depends on you, but generally 10+ hours per week?
 - Over the years, I've seen up to 100 hours (I think that's an overkill).
- **Can I add you in LinkedIn?**
 - Absolutely! Mine is <https://www.linkedin.com/in/yongwhan>.

Selected Questions from Lecture 1 Google Form (3/5)

- **Can I miss more than 2 lectures?**
 - Only in extenuating circumstances (e.g., a major health issue), but it requires an explicit written communication between you and the teaching staff.
- **Would you say interviewing in C++ is better than doing it in other languages?**
 - Not necessarily better, but sometimes required (quant).
- **How does the TechPrep AI daily challenges work?**
 - One question per day, for now, though we may consider expanding it to more. The submission window will close in rolling 24 hour window.

Selected Questions from Lecture 1 Google Form (4/5)

- Are the appointments for mock interviews and resume critique unlimited and throughout all the semester?
 - I will try my best! Traditionally, it has been!
- What is the biggest advice you would give to improving timing in interviews/contests?
 - Solve problems that are slightly above your level.
 - Solve many problems (e.g., set a goal of 10 a day as a starter).
 - Participate in the live contests and **fill in the missing knowledges!**
 - **Remember:** for tech interview, the goal is complete the weekly contest in **60 minutes** or under. For quant interview, **20 minutes**.

Selected Questions from Lecture 1 Google Form (5/5)

- **If you could teach any other course at Columbia which one would it be?**
 - I like teaching this course and competitive programming course.
 - algorithms? discrete math? The format will likely change from the current offering, to put a greater emphasis on implementations.
- **Will there be any A+ in this course?**
 - Absolutely, as you may check from the syllabus.
- **Favorite city? Favorite food? Hobby?**
 - Las Vegas or Waikiki? sushi or hot pot or barbeque? poker or travel?

Any questions, so far?

- If not, let's dive right onto technical topics! :)

Array

- Knowing array well is **crucial** for your success in any interview.
- At the very least, you should know how to use one in your favorite language.
 - For C++, use **vector**, a dynamic array.
 - Java's ArrayList vs C++'s vector vs C's array.
- It is often a **building block** for a more complicated data structure.

Array: Question

- Compute binomial coefficients $C(n,m)$ where n and m are at most 50 (Pascal's triangle).
 - Example Input #1: (5, 2)
 - Example Output #1: 10
 - Example Input #2: (2, 5)
 - Example Output #2: -1

Array: Answer

Time Complexity: $O(n^2)$

Additional Space Complexity: $O(n^2)$

```
int SIZE=55;
if(n<m) return -1;
long long a[SIZE][SIZE];
for (int i=0; i<SIZE; i++) {
    a[i][0]=a[i][i]=1;
    for (int j=1; j<i; j++)
        a[i][j]=a[i-1][j]+a[i-1][j-1];
}
return a[n][m];
```

String

- You can think of string as a “vector of characters.”

String: Question #1

- Convert well-formed string to integer.
 - Example Input #1: "123"
 - Example Output #1: 123
 - Example Input #2: "0"
 - Example Output #2: 0

String: Answer #1

- Method #1: built-in function

```
stoi(s);
```

- Method #2: stringstream

```
stringstream ss(s);  
int n; ss>>n;  
return n;
```

- There is also the usual way by inspecting s, character by character.

String: Question #2

- Check if a string is a palindrome.
 - Example Input #1: "abc"
 - Example Output #1: false
 - Example Input #2: "aa"
 - Example Output #2: true
 - Example Input #3: ""
 - Example Output #3: true

String: Answer #2

- Time Complexity: $O(n)$
- Additional Space Complexity: $O(1)$

```
int n=s.size();  
for (int i=0; i<n/2; i++)  
    if(s[i]!=s[n-1-i])  
        return false;  
return true;
```

Linked List

- Know differences between **singly** linked list vs **doubly** linked list.
- I do not like linked list but **some people love it**.
- Use templated version of node:

```
template <typename T>
struct ListNode {
    T data;
    shared_ptr<ListNode<T>> next;
}
```


Linked List: Question #1

- Search a key in the linked list.

Linked List: Answer #1

Time Complexity: $O(n)$

Additional Space Complexity: $O(1)$

```
while(L && L->data != key)
    L = L->next;
return L;
```

Linked List: Question #2

- Given the head of a singly linked list, reverse the list, and return the reversed list.

Linked List: Answer #2

Time Complexity: $O(n)$

Additional Space Complexity: $O(1)$

```
ListNode* reverseList(ListNode* head) {  
    if (head==NULL || head->next==NULL)  
        return head;  
    ListNode* p = reverseList(head->next);  
    head->next->next = head;  
    head->next = NULL;  
    return p;  
}
```

Stack and Queue

- **Stack**
 - “Last-In, First-Out” (LIFO).
 - Depth-First Search (DFS).
- **Queue**
 - “First-In, First-Out” (FIFO).
 - Breadth-First Search (BFS).
- Breadth-First Search (BFS, using queue) and Depth-First Search (DFS, using stack) will be covered towards the end of Module I.

Stack and Queue: Question

- (**max stack**) Implement a stack that in addition to pop and push, provides a max function, that returns the max element in the stack.

Stack and Queue: Answer

```
class Stack {  
    private:  
        struct data { int element, max; };  
        stack<data> stk;
```

Stack and Queue: Answer (con't)

```
public:
```

```
bool Empty() const { return stk.empty(); }
int Max() const { return stk.top().max; }
int Top() const { return stk.top().element; }
int Pop() const { if(!stk.empty()) stk.pop(); }
void Push(int x) {
    stk.emplace(data{x,
                    max(x, Empty() ? x : Max())});
}
```


THANK YOU

