DSA - 63 REVISION OUTLINE

FOR MIDTERM EXAM 2

# Fundamentals of each lesson:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Lesson** | **Contents** | **Note** |
| 1 | OOP | * Definition and example. Components of a class * OOP design principles * Overloading methods: definition and example * Inheritance: definition, example, subclass, superclass, main features | Make sure you clearly understand the concepts and explain them in your words |
| 2 | Big-O | * Worst case, Big O, time complexity, space complexity * Complexity của các method cơ bản của mỗi loại data structure, search, sort algorithms |  |
| 3 | Recursion | * Definition, two types of cases * Compare recursion & iteration (pros and cons) * Problems: Factorial, Binary search, Fibonacci, Power of number, Quick sort, Reverse string, Hanoi Tower |  |
| 4 | Array | * What are Python sequence types and efficiency of default operations in Python sequence type. * Difference between low-level arrays and referential arrays. * Definition, example and application of dynamic arrays. * Non-recursive solution**s** to Fibonacci problem | Try searching for some Fibonacci solutions which utilize arrays. |
| 5 | Stack | * Định nghĩa, cơ chế, ứng dụng * Cơ chế của các method cơ bản: isEmpty, len, push, pop, top * Parenthesis checking, reversing, sort, convert numeral system |  |
| 6 | Queue | * Định nghĩa, cơ chế, ứng dụng * Cơ chế của các method cơ bản: is\_empty, len, enqueue, dequeue, first * Queue implementation by: Array-based, Max capacity * QueueByStack, StackByQueue * Buy tickets problem |  |
| 7 | Linked list | * Definition of Singly Linked List and Doubly Linked List and their applications. * Advantages and disadvantages of Linked List over Array-Based types and vice versa. * Linked List class attributes and methods: insert, delete, sort, reverse. | Aspects to think about for methods/functions: mechanisms, efficiency |
| 8 | Tree | * Định nghĩa, ví dụ (cây thế hệ..), các terminologies và properties (node, root, child, parent, leaf, height, depth, level…) * properties của general tree, binary tree, binary search tree * Các method cơ bản của BT: add\_left, add\_right, is\_root, is\_leaf, check\_ancestor, depth, height * Linear data structures vs. non-linear data structures * find all right/left leaf nodes, search for node in BST |  |
| 9 | Tree traversal | * Cơ chế (cách hoạt động) của các loại tree traversal algorithms:   + Preorder   + Postorder   + Breadth - First   + Inorder * Applications:   + Calculate number of nodes (with given condition)   + Sum of all nodes   + Find max depth, depth of lowest leaf * Diagonal tree traversal |  |
| 10 | Priority Queue | * Definition and real-life examples of Priority Queue * Definition and structure of Binary Heap * Methods in a MinHeap class: add, remove\_min, heap\_sort algorithm, min\_heapify * Practical Problems:   + How to find the median of a data stream making use of heaps?   + How to find k smallest elements in a sorted matrix using heaps? [Link](https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/description/) |  |

## Một số câu hỏi ví dụ:

* Cho 1 ví dụ/vấn đề/bài toán cụ thể -> Hỏi output?
* Structure này sẽ ứng dụng giải quyết bài toán/vấn đề cụ thể như thế nào -> Hỏi cách làm?
* Ứng dụng thực tế (VD: Queue: danh sách chờ,...; Stack: …)
* So sánh:
  + Array vs Linked List
  + Singly Linked List vs Doubly Linked List
  + Heap vs Stack
  + …
* Pros and cons
* Dynamic structures là gì? Cho ví dụ?
* Các thuật toán sorting algorithm đã học (quick sort, insertion sort, heap sort): So sánh, giải thích cơ chế hoạt động, time complexity (nếu có hiểu biết về các loại sort khác nữa sẽ là điểm cộng: merge, selection, bubble…)
* Binary search: nắm được cách thức hoạt động, so sánh với Linear search
* Cùng 1 bài toán, so sánh các phương án giải quyết? (dùng trong tình huống nào, time/space complexity?) (VD: Bài toán Fibonacci)