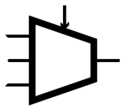


Demux DSM Batches - Curriculum

General Warm-up

1. Idea of running quantities
2. Peeking to the other side
3. Handling overflows
4. Introduction to STL
5. Memory layout of C/C++ programs

Binary Search

1. The Predicate framework
2. Intuitive Search spaces
3. Non-intuitive search spaces

2 Pointers

1. Index based
2. Window based

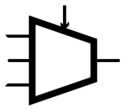
Sorting

1. Comparison based iterative sorting
    a. Applications of sorting
2. Custom comparator functions
3. Non-comparison based iterative sorting
    a. Idea of bucketing
4. Recursive sorting
    a. Divide and Conquer Paradigm
    b. The partition algorithm

Linked-lists, Stacks and Queues

1. General non-intuitive patterns involving these DS

Recursion



1. General framework for solving recursive problems
2. Different types of decisions encountered generally
3. A view of sub-problems
4. Introduction to back-tracking

Backtracking

1. Differences with Recursion
2. Classical questions and patterns involving Backtracking

Unordered Search Structures

1. Concept of Hashing
2. General Patterns in questions involving Hashing
3. Need for an order

Ordered Search Structures

1. BSTs and Heaps
2. General Patterns in questions involving ordered SS

Data-structure Design

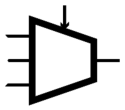
1. Use the fundamental Data-structures as building blocks to construct more advanced data-structures

Trees

1. Traversals in trees
2. General patterns in solving hard recursive questions on trees

Strings

1. Strings in STL
2. Pattern matching - KMP
3. Manacher’s algorithm
4. Tries and general patterns involving strings



Dynamic Programming

1. General Framework for DP
2. 1D DP in O(n)
3. 1D DP in O(n^2)
4. 2D DP in O(n^2)
5. 2D DP in O(n^3)
6. Matrix DP
7. Other types of decisions
8. DP on trees and graphs
9. DP and bitmasking

Graphs

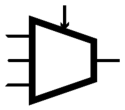
1. Representation
2. Traversals
3. Connectivity
4. Cyclicity
5. DAGs
6. Topological Sorting
7. Disjoint Data-structures: Union-find algorithm
8. Tarjan's Algorithm
9. Shortest paths
10. Spanning trees

System Design

1. Elements of Systems - CAP theorem, load balancing, sharding, etc.
2. Case studies of frequently encountered systems like Instagram, Netflix, etc.

Web-Development

1. Version Control - Git/BitBucket
2. AWS - EC2 Server, Security Groups, Load Balancer etc
3. Building a website using a front-end and a back-end framework - Frameworks TBD



4. Deploying on AWS - Full functional Website
5. Integration - DNS Config, Domain Name Config

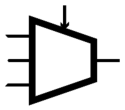
Object-oriented systems

1. Fundamentals - SQL, ACID, Normalization
2. Case studies like Splitwise, Parking-lot order, Unix file-search, etc.
3. Design Patterns

Advanced DS (if needed):

1. Segment Trees
2. Fenwick trees

Mock-interviews
Assignments and doubt classes
Resume and CV building sessions
Interview experience sessions from Alumni
Weekly hackathons



FAQs

x

- **Total program duration and timings?**

Total duration of the program is roughly 8 weeks. Sessions generally take place in the evening from 8-12 PM but can change according to availability of students over weekends.

- **Does Demux provide placement assistance/Referrals to companies?**

Because of this ongoing pandemic, most of our partner companies have stopped hiring. So we cannot promise anything.
However, we'll be having sessions on LinkedIn and profile building as well as sessions from people who got off-campus success so that you can apply off-campus. Apart from that, Demux Alums have already made it to almost every big company in India and they will keep sharing their company specific experiences Intermittently with you.