Standard Algorithms Quick Revision Using System Design

Name - Pankaj Kumar

Section - KO138

Roll Number - 05

Registration Number - 11910254

Course Name - GEN (331) WORKSHOP ON DESIGN THINKING FOR INNOVATION

Teacher Name - Respected Rameshwar Cambow

CA-2

Introduction

In this project we design a Low-Level System. Which consist most asked interview Question in Tech Company. Simply system consists all Standard Question which are frequently asked in interview. Using this system, we can revise all standard algorithms before our placement. To design the system, we used low level design (LLD).

Low-level design (LLD) is a component-level design process that follows a step-by step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. Post-build, each component is specified in detail. The LLD phase is the stage where the actual software components are designed. During the detailed phase the logical and functional design is done and the design of application structure is developed during the high-level design phase.

To design the system, we used low level design because low-level design document (LLDD) is to give the internal logical design of the actual program code. Low-level design is created based on the high-level design. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

A good low-level design document makes the program easy to develop when proper analysis is utilized to create a low-level design document. The code can then be developed directly from the low-level design document with minimal debugging and testing. Other advantages include lower cost and easier maintenance.

To revise the algorithms, we have to just execute the code in c++ compiler. Select the question which you want to revise.

For study the algorithms we have to visit the different – different website. But here you can easily excess and revise the algorithms in single place. Like if you selecting any algorithms then you have to choice which things you want about the algorithms like if you want to sudo code then , you can easily get sudo code , if you want the short description or want to see the working or want to solve the problem based on that algorithms you can easily get.

Using this system, you can quickly revise all the algorithms easily. System has user friendly interface. Like if you are making your project, in your project required any algorithms then you can directly implement these algorithms in your project. If you want modify the system then you modify the system. Simply means you can scale the system. Like if you want to add your own algorithms then you can easily add your algorithms. System architectures are simple. Anyone can understand easily.

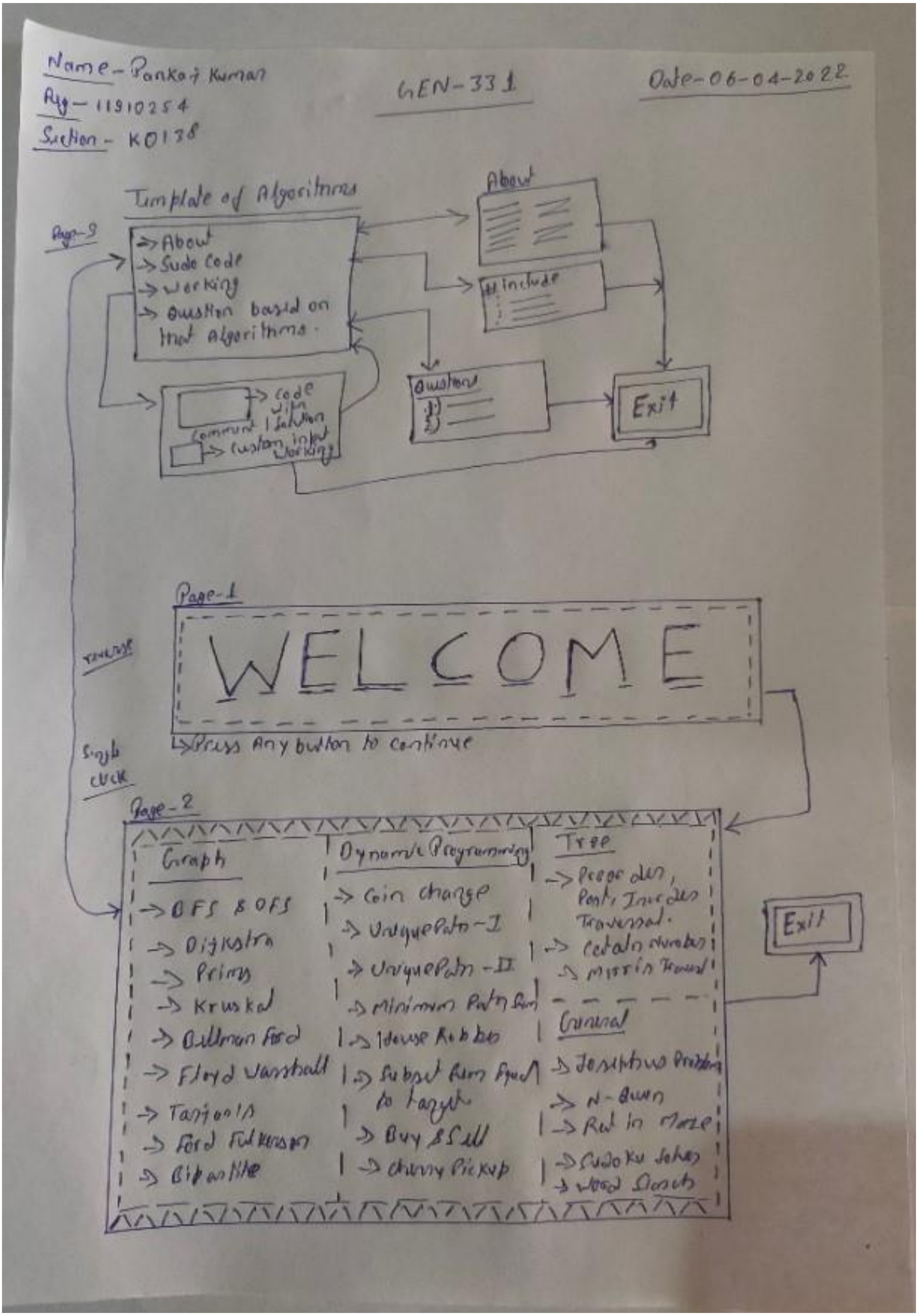
Like If you revise from system then its take very less time compare to hand written Notes. Like if you refer hand written notes then for executing the program you need to write whole program in compiler. But here all programs already running in compiler you need select only algorithms name. after selecting the algorithms, you got everything about algorithms. You can take any number of times custom input (manually taken by user) in this system. You can able see how actually program work from inside. For every algorithm you can get complete details.

For executing the algorithm, you have different options like you execute the program any operating system like – Windows, Mac, Linux. Even you can run any browser like – google chrome, Edge browser, Mozilla Firefox. Even you can also run in Android phone, IOS Phone. Just download C++ compiler or in browser just open any C++ compiler and run the program.

Whole System have less size then you easily share, to your friends. Whole system is optimized, according time complexity and space complexity. That why system take less space and less time to execute. In System every code is written in simple form, even anyone can understand the algorithms easily. System consist 50 standard question, from topic Like Graph, Dynamic Programming, Tree, some general question which are based on math.

Design and Architectures

We will design our model on below architectures.



Before Survey

Before survey I will thinking that I will add every topic of data structure. Which are important for the placement. For example -

i) Array

ii) String

iii) Stack

iv) Number Theory

v) Binary Tree

vi) Binary Search Tree

vii) Segmented Tree

viii) AVL Tree

ix) Recursion

x) Backtracking

xi) Searching & Sorting

xii) Graph

xiii) Dynamic Programming

These are topic I include in this project. From all topic I will select most Frequently asked Question Which are based on Standard algorithms. For example, if you solved Nearest Greater to Right and Left. Then you can solve the Nearest Smaller to Left and Right. If solve these Question then you can ably solve the hard question which are like Stock Span and Maximum area in Histogram easily. Which are based on same pattern. That why I include all standard algorithms which are base of most of Question. If you able understand the base question then you can solve large number of variety similar question which follow some similar pattern. For every question, I will show the working of code. Like how code is actually Working from inside like debugging. For designing the system, I will use the Linux Operating System like Centos, RedHat, Ubuntu. In which I will used c language. Also including custom input in which when user give the input, I will store the data permanently. When user closed the system then custom input given by user is save other files and user can able to use again in the model.

After Survey

After survey I got response related to data structures topic like-

Number Theory – 9.1 %

Tree Data Structure – 18.2 %

Graph Data Structure – 13.6 %

Dynamic Programming – 59.1 %

From Above survey we can see the response, student interested more on Dynamic Programming, Tree and Graph. Which most important topic of placement. Form these topics, interviewer always asked question. In online exam also 100 % probability is their tech company asked some question from Dynamic Programming and Tree Data structure. That why I decided more standard algorithms from Dynamic Programming and Tree. Graph is very important topic for placement. Like some which are asked all Question from Graph directly that why I added more question on Graph also. Before understanding the Dynamic Programming, we have known how recursion and Backtracking is working. If you are able to understand the recursion then you can easily understand the working of Dynamic Programming that why I also added some question of recursion which are recursion standard problem.

When I got the survey response then I realize that student want to clear cut code or code which can easily understandable. Some platform is their which provide the solution but it is difficult to understand that why I will try to give the best solution of every algorithms even anyone can understand easily.

Before Survey I think that i will include only working of algorithms but after survey I will include about section of code, sudo code, question based on that algorithms.

For designing the system, we will used c++ language which can run everywhere. We design in windows OS Because most of student using Windows OS. Student can able to run the model on any system means model are system independent. Even we can run any phone, in which internet is working means you can run the system on browser. Some features are removed like model cannot store the custom input permanently which are given by the user. Because it takes more space. In this project i will added documentation. Because it helps to new user to used the system in efficient manner. Apart from these users can also add their own algorithms. Can able to see the working of inside the code.

List of Algorithms

Graph

- ➔ BFS and DFS Traversal
- ➔ Topological Sort
- ➔ Detect Cycle in Directed Graph
- ➔ Detect Cycle in Undirected Graph
- ➔ Shortest Path in DAG
- ➔ Kosarajus Algorithms
- ➔ Dijkstra Algorithms
- ➔ Prims Algorithms
- ➔ Kruskal Algorithms
- ➔ BellmanFord Algorithms
- ➔ FloydWarshall Algorithms
- ➔ isBipartite Graph
- ➔ Articulation Points in Graph
- ➔ Bridge in Graph
- ➔ Tarjan's Algorithms
- ➔ Ford Fulkerson Algorithms

Dynamic Programming

- ➔ Coin Change
- ➔ Frog Jump
- ➔ Frog Jump with K^{th} Distance
- ➔ Buy and Sell
- ➔ Unique Path Sum - I
- ➔ Unique Path Sum - II
- ➔ Minimum Path Sum
- ➔ Cherry Pickup
- ➔ Longest Common Subsequence
- ➔ Longest Increasing Subsequence
- ➔ Longest Palindromic Subsequence
- ➔ House Robber
- ➔ Subset Sum Equal to Target
- ➔ Partition a Set into two Subsets Absolute Difference
- ➔ Counts Subset with K Sum

- ➔ Word Break Problem
- ➔ Egg Dropping Puzzle
- ➔ Rod Cutting
- ➔ Wildcard Matching

Tree

- ➔ Preorder, Inorder, Postorder Traversal (DFS)
- ➔ Levelorder Traversal (BFS)
- ➔ Zig Zag Traversal
- ➔ Boundary Traversal
- ➔ Morris Traversal
- ➔ Lowest Common Ancestor
- ➔ Catalan Number

General

- ➔ Josephus Using Recursion
- ➔ Generate Parentheses
- ➔ Tower of Hanoi
- ➔ Trapping Rainwater
- ➔ Job Sequencing
- ➔ N-Queen
- ➔ Rat in a Maze
- ➔ Sudoku Solver
- ➔ Word Search
- ➔ Boyer-Moore
- ➔ Floyd's Tortoise and Hare (Cycle Detection)

Advantages / Disadvantages

Advantages

->With Click you can execute the whole program.

->You can run on any operating system (any laptop or modern phone).

->You can also directly execute the program, from any web compiler.

->You can easily share the program anywhere. Because program store in single file.

->You can also add your own algorithms. Means easy to scale the system.

->You can able to take your own custom input.

->Easy to use, because user friendly interface.

->For every algorithm you got everything like- about, code, question etc.

->You can also able to modify the code according to you.

->Anyone can understand the code easily.

Disadvantages

->To revise the algorithms, we have to run the program in c++ compiler only.

->If you remove any one line then program not execute.

->You cannot execute the program in another compiler.

->For executing the program, first you have to arrange a laptop or phone.

->Unable to store your previous custom input (given by user manually).

Project Progress

“Standard Algorithms Quick Revision Using System Design” in this project I will already collect the Algorithms list. And done with algorithms solution. In this project already design the architectures of System. And also done with solution of algorithms but some algorithms I left with solution. But till next week I will be done with all solution of algorithms. Currently working on implementation part of system like how can optimize the system in better manner. Creating Command Line Interface. Which is user friendly anyone can easily use the system. Also, with working on Documentation part also. If anyone who are not able to understand the interface. He / She can easily understand the system working from documentation part. In documentation part we add everything which necessary. And every step by step we include on documentation part. We try add more features like video solution of Some Question which are hard and for practice Question we also adding some Question. If anyone who understand the algorithms and want to practice some Question on that algorithms then they can get easily question from this system without searching on google and practice to improvement. We will give user to freedom to modify the system also. In this system we add algorithms which are frequently asked in tech company. Our moto is adding good question which are really important for placements. For all the algorithms, we made the solution easy because anyone can understand easily. When we want to revise the algorithm, we have executed the system file. For interacting with interface provide special key from special key user have to write the single digit key only to execute the program not algorithms name to write, if you’re writing the name then it takes more time that why we provided a special key. These things are implementing now. Apart from this we also make simple video. Which explain how can use this project in efficient way to improve the algorithms skills. To implement the whole system, we used C++ language Because it c++ language run faster than other language like java, python, kotlin etc.

Conclusions

Let’s highlight some key features of this system.

-> System contain 50 standard Question which important for interview preparation.

-> To run the system, you have to execute the source code.

-> For every question you got – about, code, working, question based.

-> Whole Source Code is stored in single file.

-> You can run source code in c++ compiler any os, web, phone.

-> Easy to see the internal working of code.

-> Using this system, you can able to revise algorithms quickly.

-> Easy to Use the interface.

-> You can also modify the system.

These are some key features which are available in the system. So, our final project which are “Standard Algorithms Quick Revision Using System Design” contain all these features.

References

1.) https://en.wikipedia.org/wiki/Low-level_design

2.)https://www.geeksforgeeks.org/

3.)https://www.interviewbit.com/

4.)https://leetcode.com/

5.)https://cses.fi/