
VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

# School of Information Technology and Engineering

## Final Report, DECEMBER 2020

## B.Tech., Fall-2020-2021

| | |
|---|---|
| NAME | PRIYAL BHARDWAJ |
| REG. NO. | 18BIT0272 |
| COURSE CODE | EXC1011 |
| COURSE NAME | SKILLS IN COMPETITIVE CODING |

1) A brief description of all the competitive coding sessions

A lot of competitive coding sessions were held by CodeChef VIT on various problem solving and data structures and algorithms.

**Stacks:**

Session started off with the explanation of what stacks is with regards to computer science. Stack is a linear data structure that follows the Last in First Out (LIFO) model. Next the presenter explained the different conditions, empty and full – related to stack. Then we moved onto push and pop operations and how they work when paired with a full or empty stack. The discussion then moved on to the point of when stack should be used. As a segue from this point, different approaches to stack were introduced. The presenter then moved on to provide different examples of stack used in programming, starting off with simple ones. Afterwards, the time complexity concepts learned in the previous session were brought to focus. As we neared the end of the session a few more topics related to stack were discussed.

**Problem Solving and Algorithms:**

In this session some basic problem solving techniques and algorithms were explained to us. One of these methods was the Sieve of Eratosthenes. Different sieve methods were explained such as Prime Sieve and Sieve having linear time complexity. Near the end of the session Wilson's Theorem was also explained in detail. Lastly Garner's algorithm was explained.

**Hashmaps:**

In the final session, we started off with understanding what Hashmaps are. It is a data structure where we map keys to certain values. The presenter then gave us an example where we could use this concept – in finding the maximum occurrences of a letter in a string. Through various examples, we were taught the different ways of using Hashmaps. Moving on, we saw inbuilts Hashmaps in C++ and at the end we learned about iterators.

Very informative sessions held by Muskan Tewari and others.

2) A 100-150 words write up on recursion, explained, along with an example.

Recursion is a programming paradigm where we solve smaller chunks of the bigger problem until we solve the original problem completely. A recursive function is used for the recursion. A function is called recursive if it calls itself within the function definition. A recursive function should always have a base condition, or any other condition where no recursive call is made. I'll explain recursion using a common example of finding the factorial of a number.

A factorial of a number n is nothing but the result of multiplying numbers from 1 to n. Let us take an example of n = 5. We need 5 * 4 * 3 * 2 * 1 as the answer and n is decrementing itself by 1 at every step and stops at 1. So, all we need to do is take n, decrement the value of n by 1 and pass it to the function itself and multiply the return value with the current n. But n keeps decreasing and the recursion happens infinitely. This may lead to a time limit exceeded error.

We need to stop the recursion at some point which is why we need a base condition. Here the base condition tells us to stop when n = 1. Once we reach n = 1, we just return 1. Also if n = 0, we should just return 1, so lets add that to the condition. This way we can calculate the factorial of a number easily using recursion.

Below is the code for the recursive function.

```
int factorial(int n) {
        if(n == 1 || n == 0)
                return 1;
        else {
                return n * factorial(n - 1);
        }
}
```

3) A 150-200 word write up on our online webinars

**GraphQL session**

GraphQL is one of the modern methods of building and querying APIs. Join our core committee member Jugal, to help you tread your journey in GraphQL, know it's basics and increase your information manipulation and gathering efficiency.

**Serverless Framework with node.js**

But moving to serverless has a learning curve as well. You need to learn the intricacies of the platform you're using, including low-level details like format of the request input and the required shape of the response output. This can get in the way and slow your development process.
Today, I come with good news: your existing web framework tooling will work seamlessly with Serverless. In this post, I'll show you how to use the popular Node web framework Express.js to deploy a Serverless REST API. This means you can use your existing code + the vast Express.js ecosystem while still getting all the benefits of Serverless

**Gravitas 2020**

During VIT's technical fest – Gravitas, the Codechef team had organized a workshop on building functional webapps. This workshop was broken into two components, frontend and backend development. The backend part of the workshop was hosted by Yash Mehrotra, a full stack developer and a cloud enthusiast. He started off with trying to make the audience understand how backend development works through flowcharts and instructed on how to install the required software for backend development such as nodejs. After this he demonstrated various examples of simple web server-building through nodejs. With each new example he tried to explain different aspects of nodejs.