An analysis of coding styles at CODECHEF Platform

Aniket Kumar

Sridhar Chimalakonda

Venigalla Akhila Sri Manasa

cs20m002@iittp.ac.in

ch@iittp.ac.in

cs19d504@iittp.ac.in

ABSTRACT

In this article we summarize how the top 1000 users doing at codechef platform. We have selected top users according to their rating at codechef platforms. Recent developments in technology make available the cloud computing everywhere through internet. A variety of cloud-based programming platforms out there. Codechef is one of the famous competitive coding platform. The motivation of the paper is to figure out roadblocks and barrier of the competitive coding platforms. To get insights on how coder are doing, which language in demand and many more. In this we came to know short contest are prefer and it helps the coders in both ways. Also this study will throw some lights on functional vs non-functional use of programming language in competitive coding .

KEYWORDS

Codechef, functional, non-functional, cloud based platforms, coding style, web scrapping, codechef api, javascript_cache

INTRODUCTION

Currently internet services have become an important and recent developments in technology make available the Cloud computing everywhere through internet. In early days the task of programming did on offline computers where there is need to install required infrastructure and computing resources such as editors, compilers etc. Because of cloud computing all computing re- sources are available as software as service, platform as service and infrastructure as service [1] So, it is convenient to programming on any platform with out using a specific configured system. Modern

cloud based compilers enable to do programming on any machine irrespective of its configuration. Programming is a method that primes from an original design of mathematical model to executable computer programs. Software development involves activities such as analysis, developing understanding, generating algorithms, verification of requirements of algorithms including their correctness, and resource usage and execution of algorithms in a programming language. While coding most coders follow an edit-compile-test cycle [2]. This cycle occurs in IDEs. To keep with the large and quickly evolving code base and high levels of reprocess coders practice a cloud-based build system. Writing code for any problem involves understand of problem statement, understanding constraints and framing a solution. The main challenge while programming on modern online programming platforms is to meet the given constraints by resolving various test cases. Usually test cases are used to automated evaluation of code. The general criteria involved in programming over online platforms is first problem statement need to read, after reading the problem statement the programmer has given sample input and the resulting expected output.

An empirical study is really just a test that compares what we believe to what we observe. Nevertheless, such tests, when wisely constructed and executed and when used to support the scientific method, play a fundamental role in modern science. Specifically, they help us understand how and why things work, and allow us to use this understanding to materially alter our world. Coding has more significance in any domain and there is availability of a variety of online cloud-based programming platforms. Programming involves several phases such as requirements identification, design solution, coding and testing.

Research Question

RQ1 : Among top 1000 users, What is their most preferred coding contest type ?

RQ2 : Among top 1000 users, What contest pattern they follow till they reach 4 star and after that ?

RQ3 : What is the time duration of functional vs non-functional programmer to reach a certain goal ?

RQ4 : Among top 1000 users, What is the success ratio of up solve ?

RQ5 : What is the trend at Codechef for problem submission in functional programming language vs non-functional programming language?

DATASET

As the analysis is specific to a particular platform, which is not open source and also don't provide any API to get the data from them. So we find some work around/other ways to do so. We use third party API and web scrapper to collect all the required data to complete this empirical research.

1. Our first task is to collect who all are the top 1000 users. For this, we started web scrapping of this particular webpage "https://www.codechef.com/ratings/all?order=asc&page=2&sortBy=global_rank" by making a scrapper in NodeJs using javascript.

    1.1 PREPROCESSING

    After building this Scrapper, inside of it we are hitting an API to get 40 users in one go. And that is a json reply. We then preprocess it to get top 1000 user by hitting it 25 times and then process it through a parser written in python **"parser_UserName.py"**

2. Our Second task is to get all the contents in which the top 1000 performers participated since from the joining of this platform. We use a third party API "https://competitive-coding-api.herokuapp.com/api/codechef/{user_ID}" user_ID can be found through first dataset.

    2.1 PREPROCESSING

    After hitting the API we get all the information of a particular user in json format. We run a "parser_Contest.py" and several other parser code to get the required data.

3. Our third is to collect the code submitted by top 1000 performers, for this we get data directly by codechef. They provided with an API by which we can do so. But that require userID and password to authenticate. API is "https://api.codechef.com/contests/{contestCode}/problems/{problemCode}".

Note : All this data is available at https://github.com/cs20m002/ISE_termproject/tree/master/Data

EXPERIMENT IN THIS EMPIRICAL STUDY

RQ1 : Among top 1000 users, What is their most preferred coding contest type ?

In codechef, there are two of contest happens one is Long challenges and other is Short Challenges. Long challenges run for 10 days and competitive programmers gets ample amount of time to look and solve all the questions, whereas the Short Challenges run for 3 Hours that means it require more focus and high practise to solve all the questions. In general all the new competitive coder tries for long challenges as one can get enough time to think and code. But on the other hand experienced all not like that, after extracting all the contest in which he/she is participated for every user . We use a data visualisation tool to plot the graph so we can accumulate the statistics of every user to get this insights.

We parse the user#.json file for every user and try to collect all the long and short contest he/she participated on and apply mean distribution theorem to get a cumulative answer. There are noise in data so we use most common technique of noise removal like mean, most repeated etc.

Note : Implementation is there in "RQ1.py"

RQ2 : Among top 1000 users, What contest pattern they follow till they reach 4 star and after that ?

Based on the timestamp, we are trying to figure out what contest pattern they prefer. In other words, until when they reach 4 star, if they are following long contest and after that if they following short contest or vice versa, Or they there is no change in their is no change in pattern. For this to find out we will count all long and short contest before 4 star and after that and then find the ratio of long vs short. There will we two cases:

- First Ratio will be >1 OR < 1 and Second Ration is opposite
- First Ratio will be >1 OR < 1 and Second Ration is same

For the first case we count trend change and for the second not change then plot on the percentage graph to visualise the trend.

Note : Implementation is there in "RQ2.py"

RQ3 : What is the time duration of functional vs non-functional programmer to reach a certain goal ?

These below famous languages we are considering for our answer :

- Function Programming Lang : ADA, HASKELL, LISP, ErLang, KTNL

Based on the these we will first find who all those user who used to code in these functional programming lang. And will track the Timestamp, same will do for the non-functional people also and then compare them on basis of time and ease.

RQ4 : Among top 1000 users, What is the success ratio of up solve ?

In Used data we have given all successfully solved questions count, also partially solved question counts are also given. So we will find the value of these numbers ratio and will average them for result.

Note : Implementation is there in "RQ4.py"

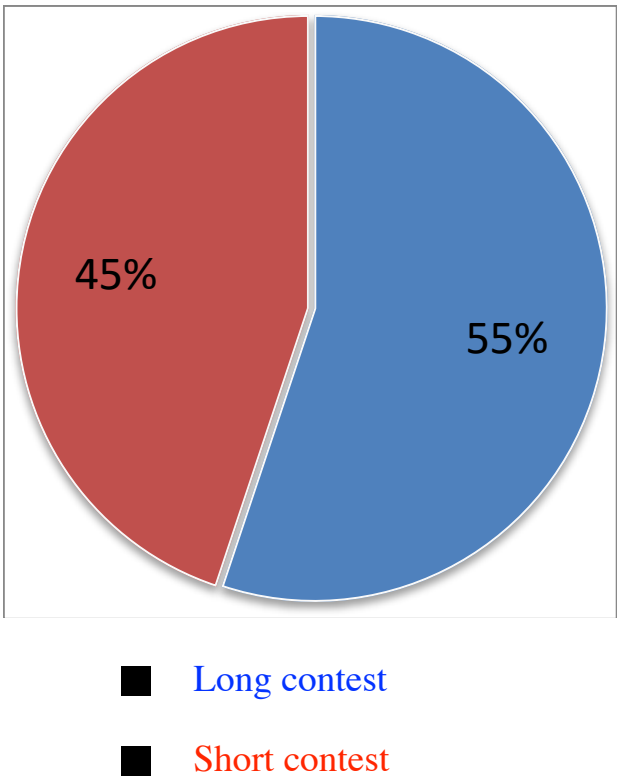
RQ5 : What is the trend at Codechef for problem submission in functional programming language vs non-functional programming language?

Don't have enough data to answer this RQ.

RESULTS

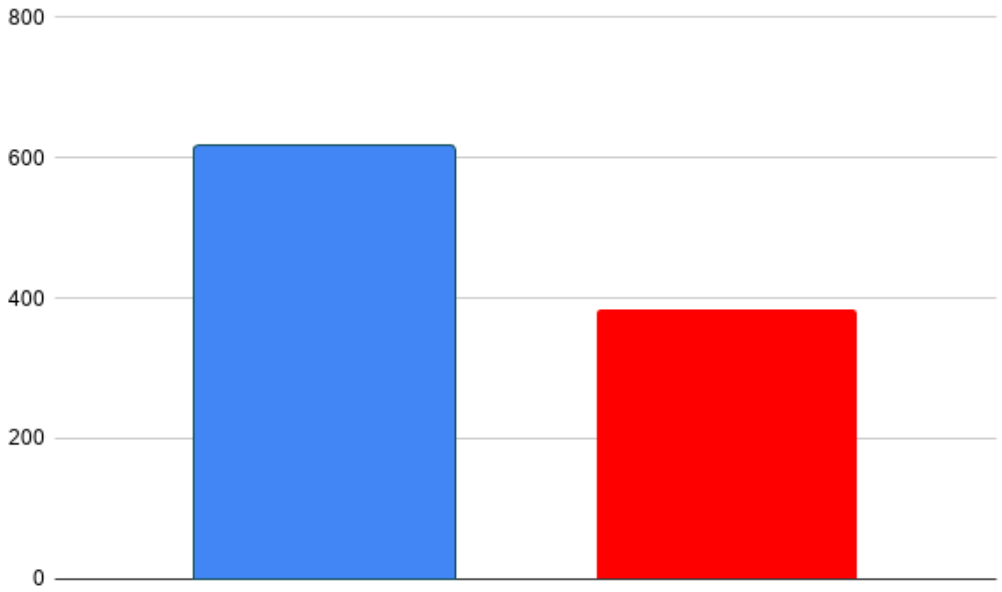
RQ1 : Among top 1000 users, What is their most preferred coding contest type ?

As we can clearly understand through this chart that most of the user go for long challenges.



RQ2 : Among top 1000 users, What contest pattern they follow till they reach 4 star and after that ?

After performing the visual analysis we came to know that out of 1000 top competitive programmer 616 follow the same trend whereas 384 follows the different trend.



RQ3 : What is the time duration of functional vs non-functional programmer to reach a certain goal ?

We haven't found any user under top 1000, who is using functional programming language for competitive coding at codechef platform. It is because due to the limitation of our dataset as all this coders are coding from 4-5 years back and at that time functional programming languages are not popular that much.

RQ4 : Among top 1000 users, What is the success ratio of up solve ?

0.07378849610489911

LIMITATION OF THE STUDY

There are many limitation of this study

- First, one can ask is why author only goes for top 1000 users. Actually, we choose first 1000 top user because they have produce a lot of to analyse. And we require a lot of data to come to a conclusion. This might be a wrong assumption as there are user with more contest but they are not in top 1000.
- Second, as we know that top 1000 user we pick is of particular day, this might be different for other day. As this ratings are very dynamic itself, but to counter this argument we spend time and analyse and finds out that there are less than 2% variation in top 1000 uses at codechef platform.
- Third, since we know it is very common that one user can code in more than one language i.e. so while analysing trend between functional and non-functional programming language there are some noise in data collection. We can be reduced by close mounting and data cleansing.

RELATED WORK

As we know cloud based services is in its early days. Codechef is a platform which is based on cloud service, so there are not much related work happens yet. But soon we have found that people are taking interest in analysing these cloud based platforms. One of these related work is where a student trying to figure out how these cloud based learning platform helps people studying in this time of pandemic.

CONCLUSION

This article talks about how coders are doing at codechef platforms. To excel their career as competitive programmer, they mainly focused on short challenges. Also when a new programmer joins the realm we saw that he first focused on long challenges, we can simply understand that in start programmers want to gain some concept and tricks to improve in this, which definitely requires time and long challenges made made for those only. Also, we figure out that there are only 1% among top 1000 users whose is using functional programming lang to solve the questions.

FUTURE WORK

One can further extend this work and might try to find out some more insights like

- Code summarisation for a particular contest, say given a question how many is doing it using one data structure and how many doing it using other, this can say which data structure is famous among users.
- Also one can do tag prediction based on the code written

ARTIFACTS

All the artifacts related to this research can be found on the GitHub repo “https://github.com/cs20m002/ISE_termproject”

REFERENCES

[1] https://www.sas.com/content/dam/SAS/sv_se/doc/Presentation/SAS -Forum-SAS-in-Cloud.pdf

[2] Hyunmin Seo, Caitlin sadowski, Sebastain Elbaum, Robert Bowdidge “Programmers’ Build Errors: A Case Study (at Google) ICSE ’14, May 31 - June 7, 2014, Hyderabad, India ACM 978-1- 4503-2756-5/14/06

[3] https://developers.codechef.com/#grant-types

[4] https://www.researchgate.net/profile/V-Kumar-21/publication/331948943_A_Survey_Paper_on_Modern_Online_Cloud-Based_Programming_Platforms/links/5fbf8f7992851c933f5d5f72/A-Survey-Paper-on-Modern-Online-Cloud-Based-Programming-Platforms.pdf