ISSN 2278-3091
Volume 13, No.1, January - February 2024
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse031312024.pdf
https://doi.org/10.30534/ijatcse/2024/031312024

Exploring the Competency of ChatGPT in Solving Competitive Programming Challenges

Md. Eusha Kadir, Tasnim Rahman*, Sourav Barman, Md. Al-Amin
Institute of Information Technology, Noakhali Science and Technology University, Noakhali, Bangladesh.
eusha@nstu.edu.bd, barman2514@student.nstu.edu.bd, alamin2514@student.nstu.edu.bd
*Corresponding Author. Email: tasnim.iit@nstu.edu.bd

Received Date : December 20, 2023 Accepted Date: January 21, 2024 Published Date: February 06, 2024

ABSTRACT

ChatGPT is an artificial intelligence-powered language processing tool that has gained widespread attention in diverse domains. It uses deep learning techniques to understand conversational context and generate human-like responses. It has also shown its capabilities in the field of computer science and software engineering by generating codes, fixing programming bugs, and explaining programming concepts. This study aims to explore the maturity of ChatGPT by assessing the correctness of solving competitive programming problems. To examine the performance of the responses generated by ChatGPT, numerous competitive programming problems have been chosen covering a variety of topics. In this study, we examine the efficacy of ChatGPT on 300 competitive programming problems spanning different difficulty levels such as Easy, Medium, and Hard. We choose an equal number of problems from each of these three difficulty levels. The performance of problem-solving using ChatGPT shows a promising result compared to the average user. ChatGPT secures an acceptance rate of 89.00%, 68.00%, and 41.00% for Easy, Medium, and Hard difficulty level problems respectively. In the overall assessment, ChatGPT achieves 66.00% acceptance rate whereas the human participants attain 52.95% acceptance rate on average. The results indicate that ChatGPT outperforms the average human problem solvers in solving competitive programming tasks. Moreover, this study also demonstrates the effectiveness of prompts and highlights the limitations of this AI tool and identifies areas where further improvements can be made.

Key words: ChatGPT, Problem-solving, Competitive programming, Programming challenges.

1. INTRODUCTION

Technology is playing a significant role in the growth of human civilization. Advancements in technology promise a better future for human beings. It has improved human life in every aspect. The evolving technology is continuously contributing to make human life easier. Advanced technologies are taking machine expertise to a higher level. Nothing seems impossible by looking at the rapid growth of technologies. The continuous evolution of technology has improved the human condition in numerous ways, offering the potential for even greater advancements in the future. The most recent popular technology that takes the technology to a greater extent is Artificial intelligence (AI). Machine learning, deep learning, artificial neural networks, and other advanced concepts and algorithms are showing promising growth. It has brought a revolutionary perception of what technology is capable of. It is being integrated with and deployed into a variety of sectors such as finance, health care, criminal justice, transportation, education, and smart cities [1]. One of the most recent and widely used tools that use AI is the Chat Generative Pre-Trained Transformer (ChatGPT). It is a chatbot that generates human-like text based on the input it receives. It is used for a variety of reasons, including content generation, tutoring and education, writing assistance, data analysis and so on.

From business sectors to medical support, ChatGPT can assist humans in numerous sectors. ChatGPT has gained popularity because of its intelligence in processing human text and communicating with helpful information. For its increasing demand and acceptance from the users, ChatGPT has drawn the attention of the researchers. Researchers are analyzing its performance in various fields ranging from comprehensive study to concise evidence that includes solving programming bugs [31], potential uses in public health [7], utility in healthcare services [17] research and practice [29], higher education assessment [5], automation of bio-informatics tasks [25], medical education and research [18], e-tourism [22] and many more. Besides that, software engineers, developers and testers take its assistance in code generation, code debugging, software testing, and even in analyzing user requirements and generating user interfaces [14]. ChatGPT has shown its promising potential in code generation and development practices for software practitioners. It has been integrated into Visual Studio Code Editor as a tool for code explanation [11]. Furthermore, it has been used as a support tool for online

behavioral task programming as reported by [3]. However, students in computer science, software engineering, information and communication technology and relevant subjects often take advantage of using ChatGPT for academic projects and assignments. Due to this, questions have arisen among the academicians regarding the potential risks and opportunities of this tool.

To understand the performance of ChatGPT, researchers employ it to answer questions from computer science, and software engineering education domains for evaluating the correctness of its responses [9, 13, 15, 16, 30, 35]. Some recent studies have paid attention to scrutinize ChatGPT's performance in answering questions typically encountered in term final examinations. These investigations aim to establish whether ChatGPT consistently and accurately addresses the typical academic challenges faced by students in these specific fields. However, the primary focus of this study is to evaluate ChatGPT's performance in tackling competitive programming challenges. To the best of our knowledge, this is one of the first investigations that has been conducted to appraise the competence of ChatGPT in solving programming challenges.

Competitive programming challenges assess not only an individual's programming skill but also the ability of critical thinking and problem-solving [20]. Competitive programming is a combination of design and implementation of algorithms. In this study, we aim to evaluate the performance of ChatGPT to address the challenges in competitive programming problems. To inspect its performance, we have collected a dataset comprising of numerous competitive programming problems covering various topics. The problems are then inquired to ChatGPT for its responses. Later the responses are submitted in one of the online judge platforms (named LeetCode) to examine and collect the verdict (e.g. Accepted, Wrong Answer). This investigation shows encouraging outcomes regarding the capabilities of this advanced AI model and also identifies the areas where further improvement is necessary.

The rest of the paper is organized as follows. The overview of ChatGPT and its evolution is discussed in Section 2. Related works from existing literature are introduced in Section 3. Section 4 describes the methodology of our investigation. We discuss the findings of this work in Section 5. Section 6 discusses the study's potential threats to validity. Finally, Section 7 concludes this work.

2. CHATGPT AND ITS EVOLUATION

ChatGPT stands for Chat Generative Pre-Trained Transformer which is a Large Language Model (LLM) based chatbot developed by OpenAI. It is a sophisticated tool that uses deep learning to comprehend and produce text that closely mimics human responses. This tool is useful for a wide range of applications across diversified fields. In its recent versions, it uses a special multi-layer transformer network architecture to process and generate human-like conversational dialogue [10]. ChatGPT has continuously evolved since 2018 through numerous updates and enhancements. It works through a transformer [33] which uses specialized algorithms that can recognize long-range patterns in sequences of data.

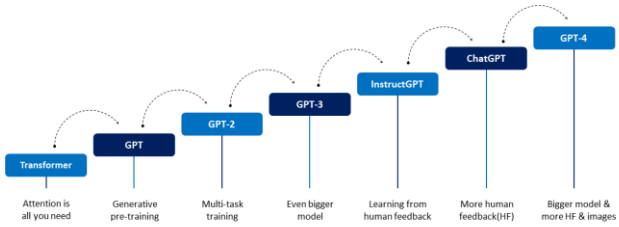


Figure 1: Evolution of ChatGPT

The transformers are good at learning from large sequences of data and can predict text, including the next word, sentence or paragraph, based on the sequence of training data. Primarily, the LLM was pre-trained on the unsupervised data to predict the next word in the sentence. In order to aid ChatGPT in analyzing conversation and developing a human-like style of response it was trained on an enormous amount of data consisting of billions of words from a wide variety of sources (e.g. books and web pages). When the GPT model was first introduced in 2018, training started with generic data and gradually progressed to data that is increasingly specialized for a given purpose. It was primarily used for language generation tasks such as text completion. However, its capabilities were limited compared to the current versions. The evolution of ChatGPT over time is depicted in Figure 1.

ChatGPT underwent a substantial enhancement in 2019, resulting in a version called GPT-2. The enhancement made in that version happened due to the expansion of the training data and the integration of fine-tuning capabilities for conversational applications [28]. In its next version (GPT-3), the training data was further extended including an increased number of parameters. The innovation of InstructGPT was a further step towards improving the model to adapt its responses to match human preferences. The innovation in this version used an additional layer of training called Reinforcement Learning with Human Feedback (RLHF) to enhance human satisfaction to the model responses [24]. The integration of human feedback enables AI systems to adapt the human preferences resulting in enhanced performance. Then a sibling model of InstructGPT named ChatGPT was trained using a greater amount of human feedback across wide range of fields.

In addition to RLHF method used in InstructGPT, this model was trained using supervised fine-tuning where human trainers engaged in conversation by playing the roles of both users and AI assistants. As a result, this generates a dataset covering a wide range of conversation scenarios which was combined with the InstructGPT dataset. Human trainers ranked multiple

model-generated responses based on factors like coherence, relevance, quality etc. These rankings were further used to develop reward-models. And these reward-models were employed to refine the InstructGPT through reinforcement learning.

Although ChatGPT is a robust language model and able to provide human-like responses, it sometimes produces biased and incorrect responses [4, 8, 32, 34]. As the model relies on a pre-trained model, it may be unaware of very recent updates of the world and fail to response properly. ChatGPT may provide outdated or incorrect information. ChatGPT often generates inappropriate responses due to the ambiguity and lack of clarity in the query. This AI tool may produce creative and imaginary responses including fabricated citations to enhance the authenticity of the generated contents. Although this tool has some limitations while replying sensitive information, ChatGPT can be a useful tool if users take exercise proper legal and ethical considerations.

3. RELATED WORKS

The introduction of ChatGPT (released in November 2022) has sparked a wave of excitement and fascination with technology. For its public availability and popularity among general users, it has also sought the attention of researchers. In this section, we will introduce several studies related to the assessments of ChatGPT in education for a variety of courses.

An investigation has been performed at Law School in University of Minnesota to evaluate the capability of the ChatGPT model in generating answers of law exam questions without any human intervention [12]. This study has explored how well AI model independently perform in examinations. The authors examined the AI generated responses and blindly assessed the responses following the school's regular practices. As a result, ChatGPT successfully passed all four courses, securing an average of C+ grade. The exam questions consisted of 95 multiple-choice questions and 12 essays. The authors explore these results extensively and highlight their effects on legal education. Authors also suggest some keywords to get better responses from ChatGPT and provide tips about the possible use of this tool in legal writing.

Another study has been conducted to evaluate the performance of ChatGPT on Multiple-Choice Question (MCQ) based exams [23]. The study encompasses an assessment of various subjects in higher education, particularly focusing on the language English. The analysis incorporates a large dataset, comprising 49,014 MCQs ranged from 53 studies and 114 question sets. The free versions of ChatGPT (e.g. GPT-3/3.5) exhibited better performance than random guessing. However, these versions typically struggled in most exams. Although the performance of ChatGPT-3.5 was worse compared to the human students, GPT-4 achieved a significant improvement and performed competitive results in comparison with human. As a conclusion of the study, authors identify ChatGPT to be a potential threat for MCQ based assessments and suggest to restrict such AI tools in the exam environment.

Pursnani, Sermet, and Demir have explored the effectiveness of using ChatGPT (GPT-4 version) by examining it for fundamental of engineering exam [26]. In this study, Pursnani *et al.* have discovered that ChatGPT pass the exam successfully, securing a satisfactory accuracy. The authors have also found that the adjustments of the prompts enhance the performance by providing more accurate responses. This research emphasizes the importance of AI related challenges in higher education and highlights the need for crafting AI-resistant exam questions to uphold the integrity of examinations. In summary, this research provides valuable insights that can be applied in the field of education.

Qadir has conducted a study to explore the potential advantages and drawbacks of ChatGPT in engineering education [27]. The author identifies some potential applications (e.g. virtual tutoring, language proficiency improvement) for students which can help them in learning. The author also expresses his doubt that the careless usage of this tool may misguide students by providing incorrect information. Although ChatGPT may generate wrong answers, it is excellent to assist in research, practise language and provide hands-on learning. Therefore, careful implementation of moral standards is important. Author suggests that clear understanding of this technology and sufficient training among the academicians will lead to take the maximum benefits by minimizing the potential risks.

A study has been conducted where authors evaluate the responses of ChatGPT for an undergraduate computer science course [9]. ChatGPT was examined in the term final question of Algorithms and Data Structures course. To mitigate the bias while evaluation, authors prepared a hand-written answer sheet from the AI generated responses and conducted blind grading. The free version of ChatGPT (GPT-3.5) achieved almost 50% marks and passed the exam narrowly. However, the premium version of ChatGPT (GPT-4) demonstrated an improvement over GPT-3.5 and secured 60% marks. This score was very close to the average performance of the 200 students who participated in the exam. The score of GPT-4 is almost similar to the average score of the 200 students participated in the exam. Finally, authors stated that ChatGPT possesses a partial understanding of computer science.

Kashefi and Mukerji have investigated the potentiality of ChatGPT to generate and debug code for numerical analysis course [19]. This investigation explores a wide range of tasks, including code generation in several programming languages (e.g. C++, Python, MATLAB), code debugging, code completion, language translation. To assess ChatGPT's

capability to recognize incomplete code, the authors consider different mathematical problems (e.g. solving linear systems of equation, finding root of a function). The authors faced several limitations, including singular matrices, reluctance to generate code, and server disconnection during generating lengthy codes. Although there were certain limitations, the authors identified ChatGPT as a proficient tool for generating numerical algorithms in different programming languages.

S. Biswas has conducted a study that was focusing on the role of ChatGPT in computer programming [6]. ChatGPT can be utilized as a beneficial tool in various areas such as code completion and correction, document generation, prediction, error fixing, optimization, text-to-code generation, and answering technical queries. Capability of providing code explanations, examples and guidance makes ChatGPT a valuable resource for technical support. Furthermore, its ability to perform programming related tasks can enhance efficiency and accuracy. In a summary, author has found ChatGPT to be a potential tool that can make a substantial impact in the field of computer programming as well as serving technical supports.

In [2], Anagnostopoulos has surveyed the impact of ChatGPT in programming education. The author has summarized positive and negative reviews regarding implementation, advantages, limitations, ethical issues, and future prospects of ChatGPT in various programming fields such as technical documentation, code management, and production, programming tutorials, and Neutral Language Processing. Several recent research articles related to programming with the assistance of ChatGPT were reviewed. The mentioned criteria related to five major questions were asked to ChatGPT itself and its responses were evaluated with research papers. Finally, author has concluded that a programmer should not solely rely on the response of ChatGPT, rather combining the programmer's knowledge and expertise can be beneficial.

Ma *et al.* [21] investigates ChatGPT’s capabilities and limitations in the field of software engineering. The study particularly focuses on ChatGPT's proficiency in comprehending code syntax, static semantic structures, and dynamic behaviors through the analysis of 2,327 code samples across nine different software engineering tasks. The findings reveal that ChatGPT has shown its ability to grasp code syntax rules and exhibit some understanding of code's static behaviors. However, it suffers from a lack of the ability to comprehend dynamic behaviors. It highlights the need for verifying ChatGPT's responses for reliability in context of software engineering.

In literature, several studies have explored the correctness of ChatGPT's responses in educational discipline. These related studies have found the potential usage of ChatGPT, accompanied by ethical guidelines for this application. However, up to our current knowledge, no investigations have distinctly focused on assessing the performance of ChatGPT in solving competitive programming challenges. Hence, this work can be considered as one of the initial endeavors to address this specific aspect.

4. STUDY SETUP

This section outlines how the study was carried out, with an emphasis on the investigation's setup. It is divided into three subsections for the ease of discussion. Initially, we describe the process of acquiring competitive programming problems along with the characteristics of these problems. Afterward, we discuss the query procedure and how the ChatGPT generated codes (responses) were collected. At last, we briefly discuss the procedure of evaluating the correctness of the generated codes.

4.1 Dataset Collection

To analyze the performance of ChatGPT in the context of solving competitive problems, we need a collection of problem statements. To collect the problem sets, we have selected an online judge platform. There are several online platforms to practice programming problems such as HackerRank, LeetCode, Codeforces, CodeChef. For this investigation, we have collected problem sets from LeetCode. We choose this platform because it offers a diverse and extensive collection of programming challenges, covering a variety of topics. Besides that, LeetCode updates its problem bank on a regular interval. It is also a popular platform for the programming practitioners.

Table 1: Summary Description of dataset comprising problem category, difficulty level, and the problem IDs in LeetCode.

Category	Difficulty	Problem ID
Array	Easy	26, 27, 66, 136, 1470, 1512, 1672, 1720, 2418
	Medium	11, 15, 36, 46, 55, 57, 63, 204, 300, 1124
	Hard	42, 149, 220, 862, 2035, 2407, 2584, 2612, 2681, 2713
Backtracking	Easy	21, 203, 206, 231, 234, 257, 342, 401, 509, 1863
	Medium	40, 79, 212, 301, 306, 638, 1219, 1849, 2002, 2305
	Hard	51, 126, 282, 679, 691, 1240, 1307, 1655, 1723, 2151
Binary Search	Easy	35, 69, 222, 278, 350, 367, 888, 1346, 1608, 2529
	Medium	33, 34, 162, 400, 456, 611, 1170, 1292, 1552, 2064
	Hard	4, 154, 327, 354, 493, 719, 878, 2286, 2659, 2790
Bit Manipulation	Easy	461, 476, 645, 762, 868, 1356, 1684, 1763, 2220, 2595
	Medium	78, 861, 1310, 1318, 1442,

		1829, 2044, 2275, 2317, 2433
	Hard	810, 1494, 1659, 1681, 1755, 1787, 1815, 1938, 1994, 2157
Dynamic Programming	Easy	70, 118, 119, 121, 338, 392, 746, 1025, 1137, 1647
	Medium	5, 198, 368, 1139, 1578, 1690, 2110, 2304, 2320, 2466
	Hard	10, 32, 44, 639, 805, 887, 913, 1388, 1977, 2338
Graph	Easy	590, 617, 700, 897, 938, 997, 1379, 1791, 2236, 2331
	Medium	547, 684, 797, 841, 959, 1557, 1584, 1615, 2477, 2685
	Hard	685, 1377, 1928, 2092, 2242, 2508, 2577, 2603, 2608, 2699
Greedy Approach	Easy	409, 455, 605, 680, 860, 942, 1221, 1323, 1403, 2578
	Medium	45, 122, 316, 402, 2202, 2601, 2611, 2616, 2645, 2680
	Hard	135, 321, 420, 1147, 1505, 1585, 2366, 2732, 2813, 2818
Heap	Easy	506, 703, 1046, 1337, 1738, 2099, 2231, 2335, 2357, 2500
	Medium	215, 264, 347, 355, 373, 378, 451, 621, 658, 659
	Hard	23, 218, 295, 407, 480, 502, 630, 632, 1606, 2642
Math	Easy	1281, 1342, 1486, 1742, 2160, 2235, 2413, 2469, 2652, 2769
	Medium	973, 1551, 1561, 1823, 1828, 2125, 2221, 2396, 2442, 2807
	Hard	60, 381, 564, 899, 1622, 1998, 2117, 2709, 2719, 2827
Sorting	Easy	1051, 1464, 1636, 1859, 1913, 2037, 2089, 2363, 2465, 2733
	Medium	1169, 1353, 1648, 1818, 2195, 2280, 2332, 2333, 2708, 2731
	Hard	76, 124, 164, 239, 664, 757, 920, 1889, 2426, 2809

LeetCode encompasses a wide range of problems covering the domains of Data Structure, Algorithm, and Mathematics. Further these problems can be categorized varying from the level of difficulty named Easy, Medium, and High. We have followed a systematic approach to collect the problem statements. Our collection consists of 300 problems gathered from ten distinct topics, with each topic comprising 30 problems. During the process of gathering problem statements, we have made sure to distribute our selection equally based on difficulty levels for each topic. Table 1 presents the problem descriptions, including their difficulty levels, problem IDs, and topics in LeetCode. The topics are listed (with brief description) as follows.

1. Array: Array is a data structure that stores a collection of elements (e.g. number, string, or other datatype) in a linear order. In LeetCode, problems categorized in this category need diverse operations of array (including manipulation, transformation, reordering, and mathematical calculations) to meet specific requirements and constraints.
2. Backtracking: Backtracking is a problem-solving technique where the objective is to find possible solution of a problem through trial-and-error. Backtracking algorithms frequently use recursion as a fundamental component of their design. Usually solving mazes or puzzles, finding permutations and combinations are the problems fall under this category. As there is a lack of problems under this category in LeetCode, we consider recursion problems under this category.
3. Binary Search: Binary search is a searching technique used to find an element in a sorted array. It works on the principle of divide and conquer where the search space is repeatedly narrowed down to half at each stage until the target element is located. This search strategy can optimize data retrieval from a large collection.
4. Bit Manipulation: Bit manipulation refers to the process of directly working with individual bits within binary representations of data. Problems under this category requires the participants to engage in bit-wise operations (such as AND, OR, XOR, shifting operations) as these operations are faster and efficient.
5. Dynamic Programming: Dynamic programming is a problem-solving strategy that involves breaking down complex problems into smaller overlapping sub-problems. To reduce the time complexity, the solutions of these sub-problems are stored in a data structure. The problems under this category are mostly optimization (either minimization or maximization) problems.
6. Greedy Approach: The greedy algorithms focus on challenges were taking locally optimum decisions at every step result in a globally optimal solution. This approach makes decisions based on what seems best at that moment, without considering whether those decisions will ultimately produce the best outcome.
7. Graph: A graph is an abstract data structure representing by a group of vertices that have data and are connected to one another through edges. This is used to depict relationships between multiple elements. The problems in this area require traversing, searching, and manipulating these kinds of interconnected structures.
8. Heap: Heap is a data structure which is used for allocating and managing memory in a dynamic manner during program execution. Problems under this category demand effective utilization of the heap data structure. The data structure is essential for several optimization problems since it provides quick insertion and extraction operations.
9. Math: Problems related to this category require mathematical concepts and techniques to be solved.

Those problems generally involve number theory, algebra, arithmetic, discrete mathematics, matrix, geometry, trigonometry, and other mathematical principles. Those problems can range from basic mathematics to more advanced mathematical concepts.

10. Sorting: The problems under this category involve the efficient implementation of different sorting algorithms maintaining specific criteria and constraints. To solve these problems efficiently, participants should have in-depth knowledge on different sorting techniques, their time and space complexity, and their applicability to the problem considering several factors like data distribution, and memory capacity.

4.2 Response Generation

To get solution of each selected problem, we have provided ChatGPT with problem statements. We have followed three approaches to present problem statements that ChatGPT can completely grasp and respond to these problem statements.

4.2.1 Declaring Problem Title using URL

The initial approach that we have adopted was to provide a specific Uniform Resource Locator (URL) that links to a problem statement on LeetCode. We also added an explicit specification of the desired programming language. We asked ChatGPT to give the solution in Java for the problem stated in https://leetcode.com/problems/<problem_title>. Here, <problem_title> was replaced with the title of a specific problem statement (e.g. longest-palindromic-substring, remove-element, sort-the-people).

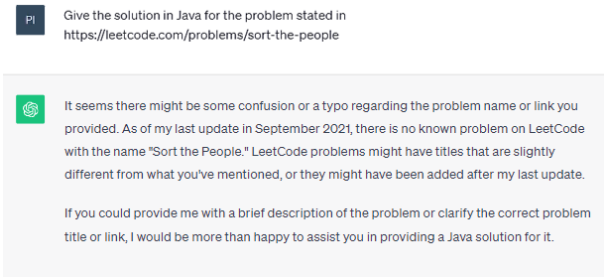
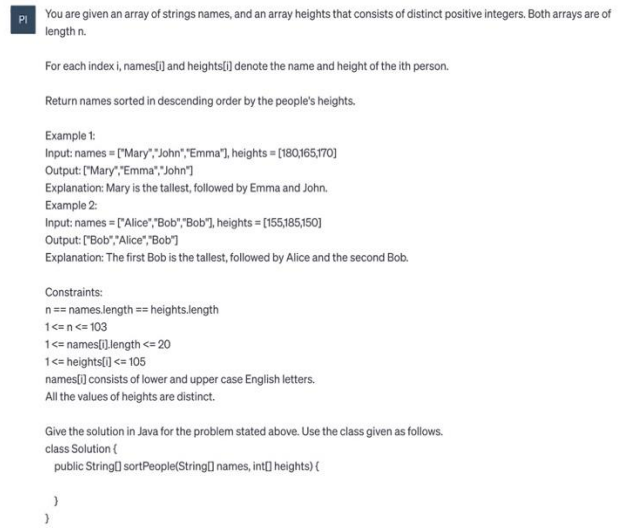


Figure 2: Sample question posed to ChatGPT and its response, describing the challenge it faced.

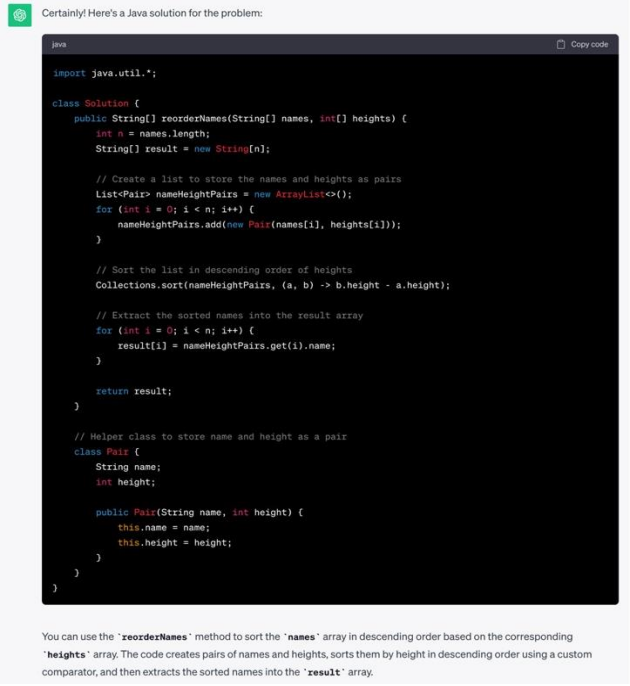
For some problem statements, ChatGPT responds well and generates code. However, this approach had faced a major challenge for several problem statements. In many cases, ChatGPT failed to retrieve problem statements, especially for those problems created after September 2021. The limitation was raised because the version of ChatGPT was last trained on data before September 2021. Figure 2 illustrates the challenge (with its response) faced by ChatGPT.

4.2.2 Replicating Problem Descriptions

To address the limitations of the previous approach, we applied another approach. In this approach, we replicated the problem description along with the test cases and constraints. We actually presented the whole problem statement as input to ChatGPT. This approach can bypass the knowledge limitation of ChatGPT to access the problem statements included after September 2021.



a) Sample Query



b) Sample Response

Figure 3: Sample query asked to ChatGPT without prompt engineering (replication problem statement) and its generated response.

ChatGPT could respond well with a generated coding solution

in many cases. However, there are some specifications (e.g. a predefined method, specific parameters, and return type) to submit the solution of a problem in LeetCode. Whenever the specifications were not met, the online judge declared the solution as a compilation error, even if the solution procedure was correct. Figure 3 illustrates an example of ChatGPT's response to a problem applying this approach.

As can be shown in Figure 3a, we have posed ChatGPT with the problem statement and asked it to provide a Java solution. Here, we have not provided ChatGPT with the other specifications of LeetCode. As a result, ChatGPT has responded with a solution (displayed in Figure 3b that did not meet the criteria required for LeetCode submission. Applying replication of problem statement approach, almost half of the solutions were not generated according to the specifications of LeetCode which is a limitation of this approach.

4.2.3 Duplicating Problem Descriptions with Prompt Engineering

As previously mentioned, the second approach frequently failed to meet the required specifications. We have applied prompt engineering and provided ChatGPT with more detailed information about a problem and its solution specification. In this approach, we have presented the code structure that is endorsed by LeetCode. This prompt enabled ChatGPT to produce a solution following the required coding structure.

Figure 4 has shown the query with necessary prompts and the response generated by ChatGPT. As depicted in Figure 4b, the solution demonstrates the ability of ChatGPT to meet the required specifications given in Figure 4a.

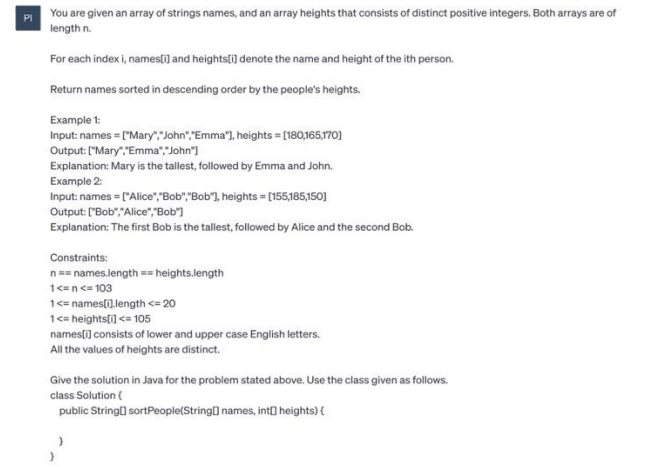
This represents how ChatGPT successfully resolved the issue of a compilation error (whereas response without prompt engineering faced a compilation error, illustrated in Figure 3b). Now we will discuss the response pattern of ChatGPT which can be divided into three distinct partitions.

- The first part comprises an introductory response from ChatGPT regarding the information it provided in the form of a solution. For example, "Certainly! Here's a solution for the problem."
- In the second part, ChatGPT provides the code (programming solution) along with inline comments to enhance the readability of the generated code. In this study, our primary focus is on this particular part.
- The last part explains the generated programming solution. This part includes data structures, variables, functions, and a brief explanation of how the process works.

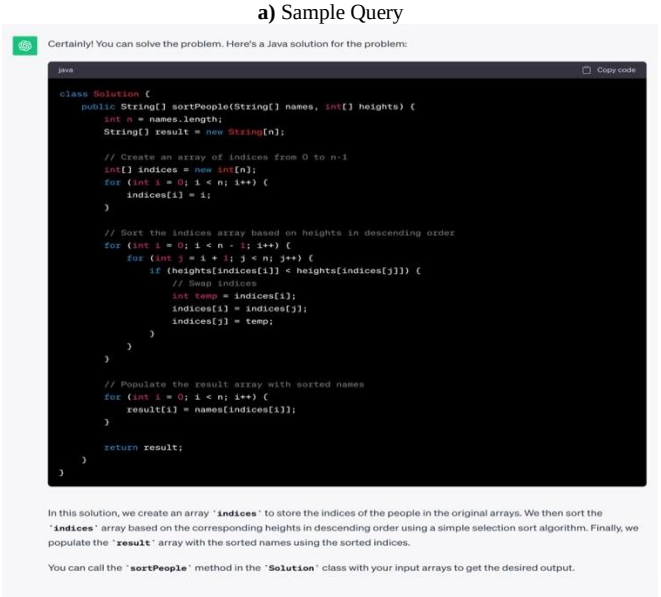
4.3 LeetCode Submission Statuses

To test the accuracy of the responses generated using ChatGPT, we have submitted the coding solutions into LeetCode. Upon submission of the solutions, we have observed different statuses of the submissions. The submission statuses are given as follows.

- Accepted: This status confirms that the provided solution is correct. It indicates that the solution passed all the test cases and does not contain any kind of issues. The solution also meets all the required specifications.
- Wrong Answer: This status indicates that the provided solution produces incorrect output for some test cases. This status underlines there are some issues in the solution. In LeetCode, the number of failure test case is also mentioned along with this verdict. For instance, the verdict can be like "Wrong answer 13/50 test cases".



a) Sample Query



b) Sample Response

Figure 4: Sample query asked to ChatGPT with prompt engineering (stating problem statement with other necessary specifications) and its generated response.

- Runtime Error: This status reveals a flaw in the solution approach which could be a logical error, input/output error, null pointer error, exception, and other similar issues. It underscores that the provided solution has improper code implementation.

- Time Limit Exceeded: It simply indicates that the provided solution takes an excessive amount of time to execute. It occurs when the provided solution exceeds the execution time permitted for the problem. It emphasizes that the solution approach misses the pre-defined time limit constraints.

- Compile Error: Compile error arises when the online judge platform is unable to compile the provided solution. Due to improper method declaration, wrong parameters, or incorrect return type, this error can be occurred.

5. RESULT DISCUSSION

In this section, we will assess the performance of ChatGPT in tackling competitive programming challenges and discuss on it. As previously detailed, we employed three approaches for generating response. Among these three approaches, we have found that the third approach (discussed in section 4.2.3) is the most effective one. In this third approach, we include prompts comprising the required coding structure, method declaration, necessary parameter lists alongside the problem statements while interacting with the AI chatbot. Adaptation of this approach has dramatically resolved the compilation errors. This underscores the value of prompt engineering for enhancing response quality and effectiveness. In the later part of this section, the responses generated by ChatGPT will refer to the responses generated using the third approach.

For performance assessment, we utilize the acceptance rate as the performance metric to measure the correctness of a solution. We have conducted a comparison between the acceptance rate of ChatGPT generated solutions and those submitted by human users. Table 2 presents the success rate of user submissions (including the standard deviation) and the acceptance rate of ChatGPT for each category and difficulty level. The overall acceptance rate of ChatGPT generated solutions is 66.00% while for submissions from humans, the acceptance rate is 52.95%. This suggests that the overall performance of ChatGPT is relatively superior compared to that of human problem solvers, as depicted in Figure 5.

When analyzing the overall performance taking difficulty levels into account, it becomes evident that there is a gradual decline in performance as the difficulty level increases. The acceptance rates obtained for easy, medium, and hard problems were 89.00%, 68.00%, and 41.00%, respectively. This similar pattern (gradual decrease) is also observed in the performance of human participants. To be specific, the average acceptance rate of human problem solvers are 62.72%, 54.04%, 36.87% for easy, medium and hard problems, respectively. From this observation, it can be stated that, similar to human participants, ChatGPT also faces challenges when dealing with more complex and difficult problems.

Table 2: Average acceptance rate of human submissions with standard deviation (SD) and ChatGPT generated solutions.

Category	Difficulty	User (Mean ± SD)	ChatGPT
Array	Easy	73.05% ± 16.70%	100.00%
	Medium	46.26% ± 14.10%	100.00%
	Hard	6.99% ± 12.20%	30.00%
	Overall	48.77% ± 23.78%	76.67%
Backtracking	Easy	59.42% ± 12.50%	80.00%
	Medium	48.49% ± 12.90%	90.00%
	Hard	44.71% ± 10.70%	50.00%
	Overall	50.87% ± 13.32%	73.33%
Binary Search	Easy	51.70% ± 12.00%	100.00%
	Medium	46.99% ± 9.90%	90.00%
	Hard	31.49% ± 9.10%	60.00%
	Overall	43.39% ± 13.38%	83.33%
Bit Manipulation	Easy	68.72% ± 11.70%	100.00%
	Medium	76.13% ± 4.10%	60.00%
	Hard	38.24% ± 8.10%	10.00%
	Overall	61.03% ± 18.63%	56.67%
Dynamic Programming	Easy	60.85% ± 9.70%	100.00%
	Medium	51.68% ± 10.50%	90.00%
	Hard	30.26% ± 8.20%	60.00%
	Overall	47.60% ± 15.99%	63.33%
Graph	Easy	77.95% ± 10.60%	80.00%
	Medium	69.58% ± 6.90%	80.00%
	Hard	34.44% ± 4.70%	20.00%
	Overall	60.66% ± 20.61	60.00%
Greedy Approach	Easy	61.47% ± 18.60%	80.00%
	Medium	42.09% ± 11.20%	40.00%
	Hard	38.25% ± 12.20%	30.00%
	Overall	47.27% ± 17.35%	50.00%
Heap	Easy	62.78% ± 10.00%	70.00%
	Medium	54.21% ± 11.30	90.00%
	Hard	48.48% ± 7.30%	80.00%
	Overall	55.16% ± 11.14%	80.00%
Math	Easy	85.33% ± 4.30%	90.00%
	Medium	80.98% ± 7.00%	70.00%
	Hard	32.95% ± 15.80%	30.00%
	Overall	65.92% ± 26.40%	63.00%
Sorting	Easy	75.92% ± 7.40%	80.00%
	Medium	27.62% ± 3.20%	20.00%
	Hard	42.87% ± 10.80%	60.00%
	Overall	48.80% ± 21.85%	53.33%

Apart from evaluating based on difficulty, the performance of ChatGPT has also been assessed on a category-by-category basis. The dataset used for this investigation include 300 problem statements ranging from ten distinct categories, each containing 10 hard, 10 medium, and 10 easy problems. Among these ten categories, ChatGPT performed exceptionally well

for the problems under the *Binary Search* category with the highest acceptance rate of 83.33%. In contrast, it obtained the lowest acceptance rate in solving problems using greedy algorithm, which stands at 50.00%. This indicates ChatGPT struggled most in Greedy Approach category. The submission status of ChatGPT generated responses across different categories are illustrated in Figure 6 which reflects that ChatGPT got the *Wrong Answer* verdict in 50.00% cases.

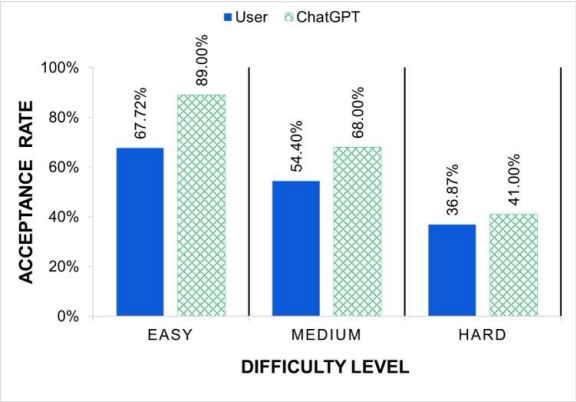


Figure 5 : Comparison between the acceptance rate of human participants and ChatGPT varying from the difficulty level of problem statements.

From Figure 6, it is observed that ChatGPT encounter the issue of *Runtime Error* most frequently (occurring in 10% cases) in handling graph related problems. Interestingly, AI-generated responses encountered no runtime issues when solving problems categorized by *Greedy Approach*, *Heap, and Sorting*. Although ChatGPT faced no runtime related issues while solving problems under *Sorting* category, these AI-generated submissions in this category faced rejection due to time constraints in many cases (particularly in 10% of cases). ChatGPT failed to meet the time limit criteria most frequently (in 13.33% times) while tackling problems categorized by *Math*.

While comparing the performance of ChatGPT with human problem solvers, we have noticed that ChatGPT demonstrates its outstanding performance in the Binary Search category, whereas human submissions exhibit the poorest performance in this category, achieving an acceptance rate of 43.39%. These result offers valuable insights into the correlations between problem categories and acceptance rates for both human submissions and ChatGPT-generated responses. Notably, fundamental and relatively easy topics such as binary search tend to receive a higher volume of submissions as these topics draw the attentions of freshers who are in the early stages of their programming journey. Therefore, the larger amount of submissions most of which are from amateur problem solvers can be a reason behind the relatively lower acceptance rate for human participants in this category compared to ChatGPT's acceptance rate.

Conversely, ChatGPT performs worse compared to average human performance in *Math* and *Bit Manipulation*. Problems under advanced topics like these tend to be mostly attempted by mid-level or expert participants who possess deep knowledge of programming and problem-solving skills. Consequently, human submissions in these categories are relatively fewer compared to the basic topics and most of them are of higher accuracy due to the expertise of the participants. Therefore, the contrast between the submission volume and accepted submissions can potentially be a reason that human acceptance rates are higher than ChatGPT in these categories. These insights suggest that continued research and further improvements are needed to bridge the performance gap of ChatGPT to solve problems under complex problem categories and enhance its understanding of complex mathematical operations.

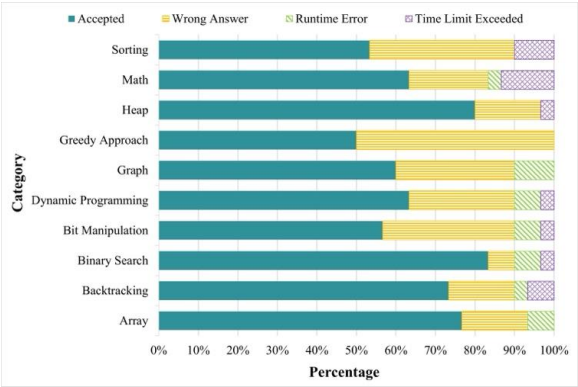


Figure 6 : Submission status of ChatGPT generated responses across different categories

6. THREATS TO VALIDITY

In this section, we will discuss a few threats to validity of this study. We have collected a dataset from LeetCode comprising of 300 competitive programming problems, covering mainly ten areas of Data Structure and Algorithm. However, more diversity and improved generalization can be attained if the dataset consists of thousands of problems, spanning almost all areas of algorithm. As we have selected the problemsets from LeetCode, we consider the submission statistics in that platform. These submission and acceptance statistics may vary for other online programming platforms (e.g. Codeforces, HackerRank, CodeChef). Therefore, it cannot be guaranteed that our findings will be universally applicable. Besides this, responses from ChatGPT were gathered from the versions that were operational in July and August. As ChatGPT is evolving continuously with new data, researchers with similar interest may suffer while reproducing the result in future.

7. CONCLUSION

Our study endeavours to shed light on the proficiency of ChatGPT in addressing competitive programming challenges. In this investigation, we examined the correctness of ChatGPT in handling programming challenges and found this

AI-powered tool as having substantial potential. We have identified the usefulness of prompt engineering while asking query to ChatGPT. From our investigations, we observed that ChatGPT outperforms human participants mostly on easy and medium level problems. However, its performance notably degrades when handling the problems characterized by Hard difficulty. The reasoning behind this outcome has been thoroughly discussed in this article. The study realized ChatGPT a prominent tool for solving programming problems and highlighted the areas where improvement is necessary. However, it also revealed its weakness in handling complex algorithms or mathematical problems. These findings may assist the problem setters of programming competitions to take an initiative to create problem statements that are more resistant for AI systems. Offering guidelines to prepare AI-resistant problem sets falls outside the scope of this study. We are planning to focus on this aspect in future work.

ACKNOWLEDGEMENT

This research is partially supported by the Research Cell, Noakhali Science and Technology University, Bangladesh, Grant No. NSTU-RC-IIT-T-23-163.

REFERENCES

1. Alattas KA, Alkaabi A, Alsaud AB (2021) **An overview of artificial general intelligence: Recent developments and future challenges.** *Journal of Computer Science* 17(4):364–370
2. Anagnostopoulos C (2023) **ChatGPT impacts in programming education: A recent literature overview that debates chatGPT responses.** *ArXiv preprint* arXiv:230912348
3. Avila-Chauvet L, Mej´ıa D, Acosta Quiroz CO (2023) **ChatGPT as a support tool for online behavioral task programming.** *SSRN preprint* SSRN:4329020
4. Bender EM, Gebru T, McMillan-Major A, et al (2021) **On the dangers of stochastic parrots: Can language models be too big?** *In: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp 610–623, https://doi.org/10.1145/3442188.3445922
5. Benuyenah V (2023) **Commentary: ChatGPT use in higher education assessment: Prospects and epistemic threats.** *Journal of Research in Innovative Teaching & Learning* 16(1):134–135
6. Biswas S (2023) **Role of ChatGPT in computer programming: ChatGPT in computer programming.** *Mesopotamian Journal of Computer Science* 2023:8–16. https://doi.org/10.58496/MJCSC/2023/002
7. Biswas SS (2023) **Role of chatGPT in public health.** *Annals of biomedical engineering* 51(5):868–869
8. Bommasani R, Hudson DA, Adeli E, et al (2021) **On the opportunities and risks of foundation models.** *ArXiv preprint* arXiv:210807258
9. Bordt S, von Luxburg U (2023) **ChatGPT participates in a computer science exam.** *ArXiv preprint* arXiv:230309461
10. Brown TB, Mann B, Ryder N, et al (2020) **Language models are few-shot learners.** *Advances in neural information processing systems* 33:1877–1901
11. Chen E, Huang R, Chen H, et al (2023) **Gptutor: A chatgpt-powered programming tool for code explanation.** *In Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky, Communications in Computer and Information Science*, vol. 1831. Springer, pp 321–327,
12. Choi JH, Hickman KE, Monahan A, et al (2023) **ChatGPT goes to law school.** *SSRN preprint* SSRN:4335905
13. Daun M, Brings J (2023) **How ChatGPT will change software engineering education.** *In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, pp 110–116, https://doi.org/10.1145/3587102.3588815
14. Fraiwan M, Khasawneh N (2023) **A review of ChatGPT applications in education, marketing, software engineering, and healthcare: Benefits, drawbacks, and research directions.** *ArXiv preprint* arXiv:230500237
15. Geng C, Zhang Y, Pientka B, et al (2023) **Can ChatGPT pass an introductory level functional language programming course?** *ArXiv preprint* arXiv:230502230 https://doi.org/10.48550/ARXIV.2305.02230
16. Jalil S, Rafi S, LaToza TD, et al (2023) **ChatGPT and software testing education: Promises & perils.** *In IEEE International Conference on Software Testing, Verification and Validation - Workshops.* pp 4130–4137, https://doi.org/10.1109/ICSTW58534.2023.00078
17. Javaid M, Haleem A, Singh RP (2023) **Chatgpt for healthcare services: An emerging stage for an innovative perspective.** *Bench Council Transactions on Benchmarks, Standards and Evaluations* 3(1):100105
18. Jeyaraman M, Priya KS, Jeyaraman N, et al (2023) **ChatGPT in medical edu- cation and research: A boon or a bane?** *Cureus* 15(8):e44316. https://doi.org/10. 7759/cureus.44316
19. Kashefi A, Mukerji T (2023) **ChatGPT for programming numerical methods.** *Journal of Machine Learning for Modeling and Computing* 4(2)
20. Laaksonen A (2020) **Guide to Competitive Programming - Learning and Improving Algorithms Through Contests, Second Edition.** *Undergraduate Topics in Computer Science*, Springer, https://doi.org/10.1007/978-3-030-39357-1 19
21. Ma W, Liu S, Wang W, et al (2023) **The scope of ChatGPT in software engineering: A thorough investigation.** *ArXiv preprint* arXiv:230512138 https://doi.org/ 10.48550/ARXIV.2305.12138

22. Mich L, Garigliano R (2023) **ChatGPT for e-tourism: a technological perspective.** *Information Technology & Tourism* pp 1–12
23. Newton PM, Xiromeriti M (2023) **ChatGPT performance on MCQ exams in higher education. A pragmatic scoping review.** *EdArXiv* https://doi.org/10.35542/osf.io/sytu3
24. Ouyang L, Wu J, Jiang X, et al (2022) **Training language models to follow instructions with human feedback.** *Advances in Neural Information Processing Systems (NeurIPS)* 35:27730–27744
25. Piccolo SR, Denny P, Luxton-Reilly A, et al (2023) **Many bioinformatics programming tasks can be automated with chatgpt.** *ArXiv preprint* arXiv:230313528
26. Pursnani V, Sermet Y, Demir I (2023) **Performance of ChatGPT on the US fundamentals of engineering exam: Comprehensive assessment of proficiency and potential implications for professional environmental engineering practice.** *ArXiv preprint* arXiv:230412198 https://doi.org/10.48550/arXiv.2304.12198
27. Qadir J (2023) **Engineering education in the era of ChatGPT: Promise and pitfalls of generative AI for education.** *In IEEE Global Engineering Education Conference (EDUCON).* IEEE, pp 1–9, https://doi.org/10.1109/EDUCON54358.2023.10125121
28. Radford A, Wu J, Child R, et al (2019) **Language models are unsupervised multitask learners.** *OpenAI blog* 1(8):9
29. Sallam M (2023) **ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns.** *Healthcare* 11(6):887
30. Sobania D, Briesch M, Hanna C, et al (2023) **An analysis of the automatic bug fixing performance of ChatGPT.** *In IEEE/ACM International Workshop on Automated Program Repair, APR@ICSE*. IEEE, pp 23–30, https://doi.org/10.1109/APR59189.2023.00012
31. Surameery NMS, Shakor MY (2023) **Use chatGPT to solve programming bugs.** *International Journal of Information Technology & Computer Engineering (IJITC)* 3(1):17–22
32. Tamkin A, Brundage M, Clark J, et al (2021) **Understanding the capabilities, limitations, and societal impact of large language models.** *ArXiv preprint* arXiv:210202503
33. Vaswani A, Shazeer N, Parmar N, et al (2017) **Attention is all you need.** *In Advances in Neural Information Processing Systems*, pp 5998–6008
34. Weidinger L, Mellor J, Rauh M, et al (2021) **Ethical and social risks of harm from language models.** *ArXiv preprint* arXiv:211204359
35. Wollowski M (2023) **Using chatgpt to produce code for a typical college-level assignment.** *AI Magazine* 44(1):129–130. https://doi.org/10.1002/AAAI.1208621