GraphArena: Benchmarking Large Language Models on Graph Computational Problems

Jianheng Tang, Qifan Zhang, Yuhan Li, Jia Li
The Hong Kong University of Science and Technology (Guangzhou)
The Hong Kong University of Science and Technology
jialee@ust.hk

Abstract

The "arms race" of Large Language Models (LLMs) demands novel, challenging, and diverse benchmarks to faithfully examine their progresses. We introduce GraphArena, a benchmarking tool designed to evaluate LLMs on graph computational problems using real-world graphs from diverse scenarios such as knowledge graphs, social networks, and molecular structures. GraphArena offers a suite of 10 computational tasks, encompassing four polynomial-time (e.g., Shortest Distance) and six NP-complete challenges (e.g., Travelling Salesman Problem). It features a rigorous evaluation framework that classifies LLM outputs as correct, suboptimal (feasible but not optimal), or hallucinatory (properly formatted but infeasible). Evaluation of 10+ leading LLMs, including GPT-4o and LLaMA3-70B-Instruct, reveals that even top-performing models struggle with larger, more complex graph problems and exhibit hallucination issues. We explore potential solutions including chain-of-thought prompting, instruction tuning, and code writing, each demonstrating unique strengths and limitations. GraphArena contributes a valuable supplement to the existing LLM benchmarks and is open-sourced at `https://github.com/squareRoot3/GraphArena`.

1 Introduction

As Large Language Models (LLMs) continue to evolve, accurately assessing their expanding capabilities becomes crucial but increasingly complex. Standardized tests, which are originally designed for humans, have become common LLM benchmarks in domains such as mathematics [34, 24, 55, 18], medicine [53, 17, 45], and multi-task collections [33, 70, 80, 98]. Despite their utility, concerns about data leakage in pretraining corpora [4, 89] cast doubt on whether LLMs are genuinely reasoning or merely memorizing. Crowd-sourced evaluations [97, 22], while reflective of real-world performance, are expensive and time-consuming [15]. Alternatively, evaluations using synthetic data like theorem proving [78, 86, 36], are rigorous but often lack real-world relevance.

To balance factors like cost, novelty, complexity, diversity, and practical applicability in LLM evaluation, the community is expanding its attention to modalities that go beyond text to include vision [55, 29], audio [91], tabular [71], and spatiotemporal [54] data. Among these, *graphs* stand out as a particularly promising avenue. Ubiquitous in many modern applications, graphs provide a robust framework for testing LLMs' capabilities on interpret relational information, process non-sequential data, and generalize across diverse structures [49, 40, 65, 81, 20, 21]. Specifically, *graph computational problems*—which typically require systematic traversal or search algorithms to solve—serve as an excellent testbed for LLMs [27, 81]. Consider the task of finding the shortest path: a model must understand the entire graph structure, identify relevant nodes, logically deduce multiple steps to trace paths between nodes, and perform calculations to arrive at a correct conclusion.

Over the past year, several studies have assessed LLM performance on graph computational problems, including notable efforts like NLGraph [81], GraphQA [27], VisionGraph [50], GITA [85], and GraphInstruct [19]. While these studies have provided valuable insights, we identify three significant shortcomings in the existing benchmarks. First, they rely on synthetic graphs generated from models such as Erdős-Rényi [26] and Barabási-Albert [5], which may not accurately reflect real-world diversity. Second, these benchmarks focus predominantly on elementary graph tasks involving small-scale graphs, limiting tasks to simple queries and counts of basic elements like nodes, edges, triangles, and rings. More complex and challenging graph problems are largely overlooked. Third, they only require LLMs to provide straightforward answers—such as *yes*, *no*, or a numerical value—rather than detailed paths or reasoning processes. This format may be utilized by models through guesswork rather than through real comprehension of the underlying logic.

In this paper, we introduce GraphArena, a benchmarking tool designed to assess the reasoning capabilities of LLMs on addressing graph computational problems. Recognizing the limitations of existing benchmarks, GraphArena incorporates three core principles:

- **Realistic Graph Collection.** Each problem in GraphArena features a subgraph containing up to 50 nodes, sampled from a rich assortment of real-world graphs, including knowledge graphs, social networks, and molecular structures. Our sampling process is designed to preserve the original graph topology and attributes, ensuring the subgraphs maintain the characteristics of their source networks.
- **Curated Task Selection.** Our benchmark offers a diverse and challenging set of tasks that examine a broad spectrum of LLM reasoning skills. Illustrated in Figure 1, GraphArena includes 4 polynomial-time and 6 NP-complete challenges, which test both intuitive (System 1) and deliberative (System 2) cognitive processes [41]. These tasks require skills from simple computations to complex multi-step analytics, long-range location identification, and heuristic approximations.
- **Rigorous Evaluation Framework.** GraphArena introduces a task-specific, fine-grained evaluation protocol to increase the rigor of assessments and reduce reliance on superficial pattern recognition. LLMs are required to identify the specific component or path in the graph that leads to the solution. Responses are then classified as correct, suboptimal (feasible but not optimal), hallucinatory (properly formatted but infeasible), or missing, enabling a nuanced comparison among LLMs.

Our comprehensive assessment of ten popular LLMs across 10,000 problems in GraphArena reveals substantial insights. Consistent with common perception, GPT-4o [2] and Llama3-70b-Instruct [3] stand out as the best closed-source and open-source models, respectively. However, even these leading models face challenges with NP-complete tasks and exhibit a propensity for hallucination, in which they frequently generate structurally correct but contextually infeasible responses. The hallucination issue is more severe on larger graphs, problems requiring more reasoning steps, and models with fewer parameters. To address these challenges, we explore three potential solutions: chain-of-thought prompting, instruction tuning, and code writing. Our findings reveal that instruction tuning and code writing offer complementary benefits in reducing hallucinations for smaller and larger graphs, respectively, though each approach has its limitations. These insights highlight that graph computational problems is a valuable testbed for evaluating and advancing LLMs towards the artificial general intelligence with strong reasoning capacities.

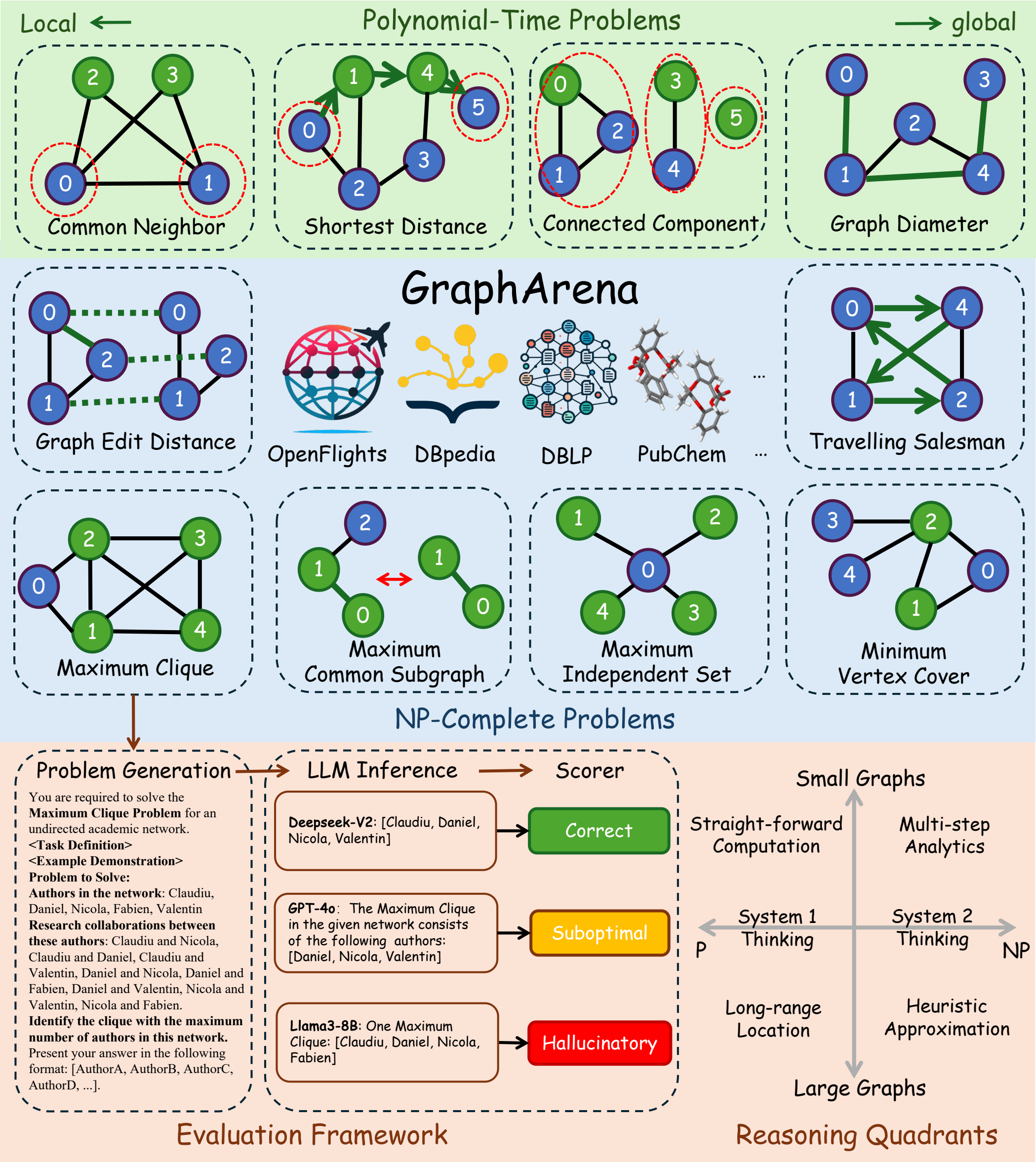
2 Benchmark Construction

Figure 1 provides an overview of our proposed GraphArena benchmark. In this section, we first discuss the collection and sampling methodologies for real-world graphs in Section 2.1, then offers detailed descriptions of each task within GraphArena and outlines the problem generation process in Section 2.2. Finally, Section 2.3 details the evaluation framework used to score the responses.

2.1 Dataset Collection

GraphArena distinguishes itself from previous benchmarks by utilizing real-world graphs instead of synthetic ones, offering a richer diversity of data. We collect graphs from the following five sources:

- **DBLP [44]** is an academic database containing over 7 million publications under the CC0 1.0 license. We use an undirected research collaboration graph for the period 2003-2018 from DBLP



You are required to solve the **Maximum Clique Problem** for an undirected academic network.
<Task Definition>
<Example Demonstration>
Problem to Solve:
Authors in the network: Claudiu, Daniel, Nicola, Fabien, Valentin
Research collaborations between these authors: Claudiu and Nicola, Claudiu and Daniel, Claudiu and Valentin, Daniel and Nicola, Daniel and Fabien, Daniel and Valentin, Nicola and Valentin, Nicola and Fabien.
Identify the clique with the maximum number of authors in this network.
Present your answer in the following format: [AuthorA, AuthorB, AuthorC, AuthorD, ...].

Deepseek-V2: [Claudiu, Daniel, Nicola, Valentin]

GPT-4o: The Maximum Clique in the given network consists of the following authors: [Daniel, Nicola, Valentin]

Llama3-8B: One Maximum Clique: [Claudiu, Daniel, Nicola, Fabien]

Figure 1: Overview of the GraphArena benchmark.

[77]. This graph consists of 1,354,852 nodes and 5,129,047 edges, with each node representing an author and each edge reflecting a collaboration that has occurred at least five times.

- **Social Network [68]** is a repository comprising over 70 different social networks under the CC BY-SA 3.0 License. We combine graphs from four large networks—Digg, Flixster, Lastfm, and Pokec—to form a dataset of 6,118,793 nodes and 40,647,227 edges. Each node symbolizes a user, and each edge denotes a friendship connection. All user names are pseudonymized for privacy.
- **DBpedia [12]** extracts structured content from Wikipedia to form a comprehensive knowledge graph. We use DBpedia1M [87], a subset of the English version, which comprises 1,160,306 nodes and 7,214,631 edges. Each node represents an entity, and each edge denotes a relationship, with all original textual content retained as attributes. It is under the CC BY-SA 3.0 License.
- **Openflights [61]** provides data for over 10,000 airports and their connecting flight routes. We construct an undirected graph where nodes are airports and edges are flight routes, weighted by the geographical distance between airports. We preserve the largest connected component in this graph, which consists of 3,390 nodes and 19,166 edges. It is under the DbCL v1.0 License.
- **PubChemQC [59]** offers a quantum chemistry database detailing ground-state electronic structures. From the PCQM4Mv2 Dataset—part of the OGB Large-Scale Challenge [35] collected from

PubChemQC—we incorporate 3,746,620 individual graphs. Each graph represents a molecule, where nodes are atoms and edges are chemical bonds, with atom types preserved as node attributes. It is under the CC BY 4.0 License.

We employ a random walk with restart strategy initiating from randomly selected nodes to effectively sample subgraphs. This technique enables us to identify the top-k most frequently visited nodes, forming local dense subgraphs that closely maintain the topological features of the original graphs. For the PubChemQC dataset, we sample molecular graphs directly by specified sizes. All graphs are treated as unweighted and undirected, except for OpenFlights which is weighted.

2.2 Task Design

Wikipedia lists 73 graph computational problems in graph theory, not counting variants. Selecting an appropriate subset for benchmarking is a notable challenge. Previous benchmarks often focused on simpler tasks like querying the existence and counts of basic graph elements such as nodes, edges, triangles, and rings [81, 27]. In contrast, we aim at incorporating more complex problems that demand advanced reasoning abilities.

To establish a nuanced evaluation of responses, we decompose each task into two requirements: satisfying the first requirement indicates a feasible solution, whereas fulfilling both requirements denotes a correct solution. We have chosen four polynomial-time tasks that require increasingly thorough understandings of graph structures:

- **Common Neighbor:** For a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with vertices $v_1, v_2 \in \mathcal{V}$, the task involves identifying the subset $S \subseteq \mathcal{V}$ where each $u \in S$ is connected to both v_1 and v_2. The objectives are to (1) identify these common neighbors, and (2) maximize their count. We utilize the DBLP academic collaboration graph for this task.
- **Shortest Distance:** In a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, determine the shortest path from node v_1 to v_2. The goals are to (1) find a viable path $(v_1, u_1, \ldots, v_2)$, and (2) minimize its hop count. We apply this task to the DBpedia knowledge graph.
- **Connected Component:** In this task, the model identifies one representative node from each connected component within a graph. For a given graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the objectives are to (1) identify a set of vertices such that each is from a distinct connected component, and (2) ensure that these vertices represent all possible connected components of the graph. This task is applied to the Social Network dataset.
- **Graph Diameter:** The diameter of a graph is the maximum distance between any pair of nodes in the graph. Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, this task entails finding (1) a shortest path between two arbitrary vertices and (2) ensuring it is the maximum possible among all shortest paths. This is performed using the DBpedia knowledge graph.

Each task is briefly described by a keyword (Neighbor, Distance, Component, Diameter) to facilitate easy reference. For the NP-complete challenges, we have selected six representative tasks:

- **Maximum Clique Problem (MCP):** Identify the largest complete subgraph, or clique, in a given graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. The task entails (1) finding a clique $\mathcal{C} \subseteq \mathcal{V}$ that is feasible within the graph, and (2) ensuring that $\mathcal{C}$ is the largest among all possible cliques. This is implemented using the DBLP dataset.
- **Maximum Independent Set (MIS):** Select the largest set of mutually nonadjacent nodes from a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. The task is to (1) identify an independent set $\mathcal{S} \subseteq \mathcal{V}$, and (2) ensure that $\mathcal{S}$ is the largest among all feasible sets. This task is performed using the Social Network dataset.
- The descriptions for the remaining four tasks—**Minimum Vertex Cover (MVC)**, **Maximum Common Subgraph (MCS)**, **Graph Edit Distance (GED)**, and **Traveling Salesman Problem (TSP)**—are provided in Appendix A due to space constraints.

Problem Generation: For each of the 10 tasks, we randomly sample 500 small and 500 large graphs to create two distinct subsets, yielding a total of 10,000 graphs. The size ranges for these graphs are tailored to the complexity of the tasks. For the Neighbor and Distance tasks, small graphs contain 4 to 19 nodes, while large graphs contain 20 to 50 nodes. For Component, Diameter, MCP, MIS, and MVC, small graphs have 4 to 14 nodes, and large graphs range from 15 to 30 nodes. For MCS, GED and TSP, small graphs have 4 to 9 nodes and large graphs range from 10 to 20 nodes. The ground truth of each problem is generated by corresponding exact graph algorithms.

After sampling graphs with the given size, we encode graph problem into text based on templates, a common practice in previous benchmarks [81, 19]. Such an example is in the bottom left of Figure 1. The process begins with an introduction and definition of the task, followed by a randomly selected example with the correct answer to demonstrate the problem-solving process. Subsequently, the actual problem to be solved is listed, including detailed graph information and task requirements. While the graphs in each problem may not appear large, their text representation often results in problems containing up to 6,000 tokens, posing a long context challenge. Detailed statistics on graph sizes and problem lengths, along with examples for each task, are provided in Appendix A.

2.3 Evaluation Framework

Unlike previous benchmarks that only verify responses as yes, no, or a numeric value, we develop a more detailed evaluation framework to prevent pattern-based guesses. For each task, LLMs are required to output *the relevant graph component or the path that contributes to the final answer*, as indicated by the green sections in each graph in Figure 1. Responses are categorized into four types through a checker: (1) **Correct:** The solution satisfies both specified task requirements; (2) **Suboptimal:** The solution satisfies the first requirement but not the second; (3) **Hallucinatory:** The solution is in the correct format but does not satisfy either requirement; (4) **Missing:** The LLM fails to provide a solution in the correct format.

For example, in the bottom middle of Figure 1, Deepseek-V2 correctly identifies the maximum clique of four authors. In contrast, GPT-4o finds a feasible but suboptimal solution with three nodes, while Llama3-8b produces a hallucinatory response, erroneously linking Claudiu and Fabien—a connection that does not exist. If we merely report numbers as in existing graph problem benchmarks, this error would be misclassified as a correct answer.

Metrics. Given the rarity of missing results in our experiments (less than 1% in most cases), our primary metrics focus on **Accuracy** (the proportion of correct answers), **Feasibility** (the proportion of both correct and sub-optimal answers), and **Hallucination** (the proportion of hallucinatory responses). We also introduce ranking-based metrics for additional comparison, particularly useful when most models yield suboptimal solutions. These metrics include Mean Reciprocal Rank (**MRR**), **Top-1** Probability, and **Top-3** Probability.

3 Experiments

Experimental Setup. In our experiments, we evaluated ten popular LLMs in GraphArena, encompassing both closed-source and open-source models. This included single models and mix-of-experts with diverse scales. Among the closed-source models, we tested the latest GPT-4o (gpt-4o-2024-05-13) [60], the well-known GPT-3.5 (gpt-3.5-turbo-0125) [62], and the cost-effective Claude3-haiku [6]. In the open-source category, our assessments included the Llama series (Llama3-70b-Instruct and Llama3-8b-Instruct) from Meta, the Qwen series (Qwen1.5-72b-Chat and Qwen1.5-8b-Chat) from AliCloud, and Gemma-7b from Google Cloud [1]. Additionally, we evaluated two mix-of-expert LLMs: Deepseek-V2 [11] with 230 billion parameters (21 billion active) and Mixtral-7x8b [39] featuring 47 billion parameters (13 billion active).

For closed-source LLMs and open-source LLMs larger than 10 billion parameters, we utilized services from cloud providers, including OpenAI, Claude, AliCloud [25], Deepseek [66], and AIMLAPI [7]. For smaller models with fewer than 10 billion parameters, inferences were conducted on our local machines equipped with four NVIDIA H800 PCIe 80GB GPUs. EGiven the high costs and time-intensive nature of evaluating LLMs, we limited our runs to a single instance per model across 10,000 problems in GraphArena. To encourage consistency in responses, we used a low temperature setting of 0.1 and imposed no constraints on the output length of any model. All responses were documented for public reference.

3.1 Main Results

In Table 1, we present the average performance of ten LLMs across four polynomial-time and six NP-complete tasks, assessed on both small and large graphs. As expected, GPT-4o consistently ranks as the top performer across most metrics and settings, with Llama3-70b closely following as the premier open-source model. Performance generally declines for all LLMs as graph sizes increase

Table 1: Average rankings and performance of 10 LLMs on 4 Polynomial-time and 6 NP-complete tasks across small and large graphs. All metrics excluding MRR are scaled to [0, 100]. **Acc.**, **Fea.**, and **Hallu.** represent Accuracy, Feasibility, and Hallucination probability, respectively. For all metrics except Hallucination, higher values indicate better performance. The best-performing open and closed-source models are highlighted in **bold**.

	Polynomial-Time Tasks (Small)						**Polynomial-Time Tasks (Large)**					
LLM	MRR	Top-1	Top-3	Acc.	Fea.	Hallu.	MRR	Top-1	Top-3	Acc.	Fea.	Hallu.
GPT-4o	**0.82**	**78.00**	**81.53**	**71.90**	**82.30**	**17.45**	**0.63**	**55.95**	**63.29**	**40.65**	**59.00**	**39.55**
GPT-3.5	0.46	36.58	42.66	34.00	52.75	47.25	0.30	18.37	25.82	14.20	30.30	69.55
Claude-haiku	0.53	45.51	49.08	43.10	53.90	46.00	0.34	23.89	29.63	18.85	33.40	66.50
Llama3-70b	**0.72**	**66.42**	**70.80**	**61.20**	**72.70**	**27.30**	**0.54**	**45.58**	**53.93**	**31.55**	**50.95**	**49.05**
Llama3-8b	0.40	30.74	34.98	28.55	45.30	53.90	0.23	12.45	15.85	9.40	19.65	78.20
Qwen1.5-72b	0.50	41.62	46.87	39.05	58.05	41.80	0.34	24.51	28.14	20.45	28.85	70.80
Qwen1.5-7b	0.23	12.79	14.67	11.85	20.75	79.10	0.16	6.11	7.27	4.60	8.40	91.20
Deepseek-V2	0.63	56.11	62.53	51.35	64.65	34.95	0.44	34.50	41.59	24.70	38.00	61.15
Mixtral-7x8b	0.48	39.81	45.04	37.80	53.00	46.35	0.31	19.24	28.00	13.85	31.75	67.20
Gemma-7b	0.35	26.29	29.39	25.15	38.60	61.40	0.20	10.31	12.21	9.20	16.05	83.95
	NP-Complete Tasks (Small)						**NP-Complete Tasks (Large)**					
GPT-4o	**0.62**	**52.36**	**62.84**	**40.83**	74.20	23.23	**0.42**	**28.68**	**43.28**	**5.93**	**49.17**	**45.60**
GPT-3.5	0.47	33.28	46.91	25.17	73.20	26.33	0.26	11.32	22.81	2.27	38.57	59.77
Claude-haiku	0.56	43.50	58.09	33.47	**78.00**	**21.10**	0.33	17.98	34.67	3.93	40.90	58.13
Llama3-70b	**0.58**	**47.50**	**60.47**	**36.77**	**75.47**	**24.53**	**0.37**	**21.01**	**40.22**	4.80	**48.77**	**50.07**
Llama3-8b	0.37	25.15	33.40	20.17	58.07	41.73	0.22	7.11	21.39	1.37	31.17	68.77
Qwen1.5-72b	0.48	36.25	48.14	28.30	69.73	27.40	0.30	13.62	31.67	3.93	47.97	51.63
Qwen1.5-7b	0.28	16.00	22.67	12.70	50.17	49.63	0.18	4.85	12.43	1.13	25.63	73.83
Deepseek-V2	0.56	43.48	58.68	33.70	74.17	24.47	0.35	19.80	36.68	**5.43**	45.00	54.97
Mixtral-7x8b	0.33	19.76	28.15	15.57	59.43	37.43	0.21	7.99	15.52	2.13	24.20	75.50
Gemma-7b	0.31	16.95	26.69	12.93	57.13	42.60	0.21	7.17	16.20	1.67	29.40	64.63

and tasks transition from polynomial-time to NP-complete, expanding the search space significantly. GPT-4o, despite its capabilities, struggles notably with NP-complete tasks on large graphs, achieving only a 5.93% accuracy rate. It indicates the most challenging scenario in GraphArena is still much larger than current LLMs' upper bound.

GraphArena also reveals a significant performance gap between LLMs with different parameter numbers. For example, in polynomial-time tasks, the disparity between Llama3-70b and Llama3-8b is notable—61.2% vs. 28.6% on small graphs and 31.6% vs. 9.4% on large graphs. This gap is significantly wider than in other benchmarks, such as GSM8K (93.0% vs. 79.6%) and GPQA (39.5% vs. 34.2%). A similar trend is observed in the Qwen series. Moreover, models with fewer parameters are more prone to hallucination, with ratios of 78.2% for Llama3-8b, 91.2% for Qwen1.5-7b, and 84.0% for Gemma-7b, indicating that GraphArena's challenges necessitate more advanced reasoning capabilities that larger LLMs are better equipped to manage.

To closely examine model performance on individual tasks, we compare the feasibility and accuracy of five selected LLMs on each task in Figure 2. Results for the remaining five LLMs are demonstrated in Figure 6 in Appendix B. Model performance varies substantially for each task, with tasks such as Diameter, MCS, and MVC demonstrate notably low feasibility (which means high hallucination ratio). This is attributed to their complex verification processes: Diameter verification requires solving a shortest path problem, while MCS verification involves graph isomorphism.

Hallucination Impact. We illustrate the hallucination probability against varying node sizes for three selected tasks in Figure 3. Results for the remaining seven tasks are presented in Figures 7 and 8 in Appendix B. These plots reveal a significant, nearly monotonic increase in hallucination ratio as node size grows from 5 to 30. For example, in the Diameter task, GPT-4o's hallucination probability rises dramatically from 18% at a node size of 5 to 80% at a node size of 30. This trend underscores that larger graph sizes, and their associated challenges in long-range multi-step reasoning, are major contributors to hallucination.

Comparison with Graph Algorithms. In GraphArena, the ground truth for each problem is computed using exact graph algorithms, and the accuracy metric evaluates LLMs' ability to match the performance of these algorithms. However, for some NP-hard tasks like TSP, GED, and MCS, exact algorithms can take hours to find the optimal solution. Therefore, we further compare three

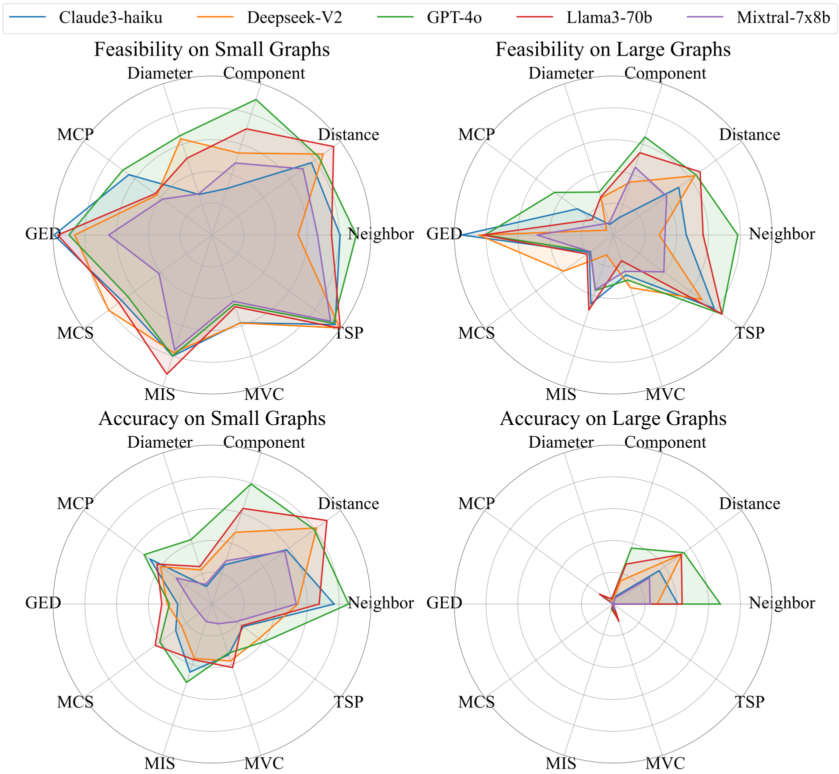


Figure 2: Feasibility and accuracy comparison of five selected LLMs on each individual task. The circles represent performance levels, progressing outward from 20% to 100% in increments of 20%.

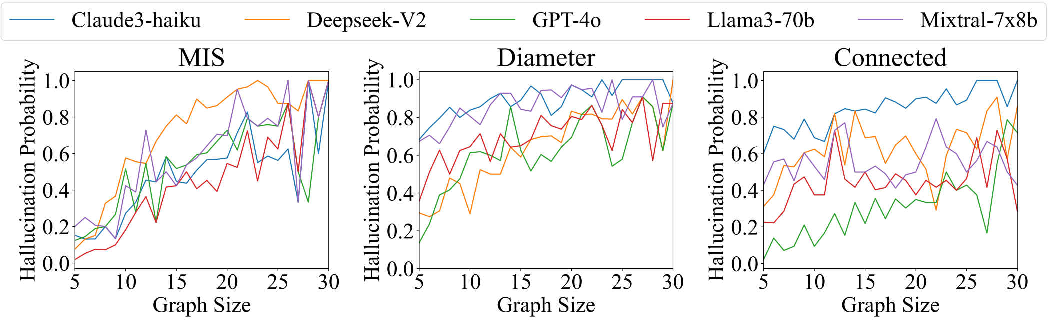


Figure 3: The influence of graph size on hallucination probability for the Maximum Independent Set, Graph Diameter, and Connected Component tasks.

suboptimal yet more efficient graph algorithms: (1) Random algorithms: randomly generate a feasible solution; (2) Greedy algorithms: choose the best option at each step; and (3) Approximated algorithms: advanced graph algorithms with problem-specific heuristics or approximations that typically better than the greedy algorithm [23, 92, 13, 10].

Figure 4 illustrates the comparative performance of GPT-4o against Random, Greedy, and Approximation algorithms on four selected NP-complete tasks, showing the percentage of problems where GPT-4o wins, ties, or loses. Comprehensive results for all LLMs are summarized in Table 4 in Appendix B. Our analysis reveals that GPT-4o consistently outperforms random solutions across most scenarios. When compared to greedy algorithms, GPT-4o demonstrates comparable performance on small-scale graphs but struggles with larger, more complex structures. Although GPT-4o rarely surpasses advanced approximation algorithms, its occasional successes hint at the potential of LLMs as alternative heuristics for NP-complete tasks.

Additionally, we conducted a preliminary comparison with three types of graph neural networks (GNNs)—GIN [88], GAT [79], and GraphSAGE [32]—as detailed in Table 5 in Appendix B. Our findings indicate that GPT-4o generally outperforms these GNNs when they lack task-specific

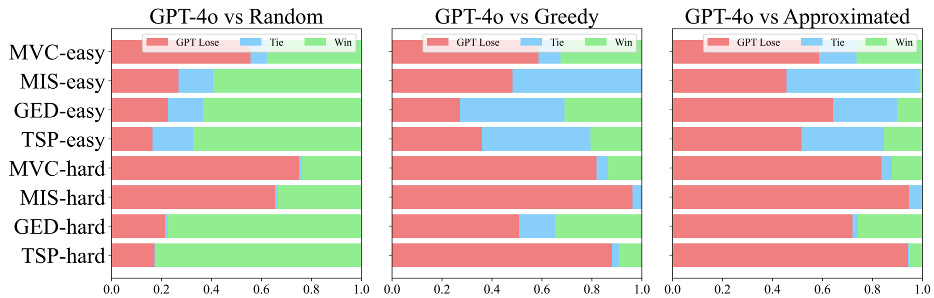


Figure 4: The percentage of problems where **GPT-4o wins, ties, or loses** against Random, Greedy, and Approximated algorithms.

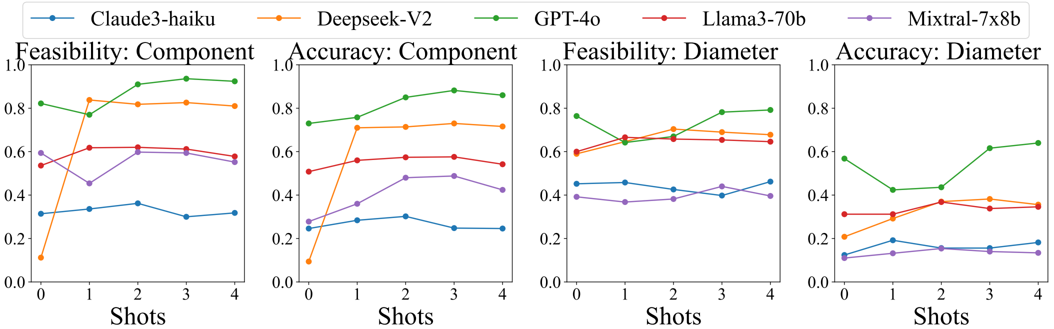


Figure 5: The influence of chain-of-thoughts with k-shot step-by-step demonstrations on the feasibility and accuracy of model performance for the Graph Diameter and Connected Component tasks.

Table 2: Performance comparison of base LLMs, LLMs finetuned on graph problems (-SFT), and LLMs prompted to write and execute code (-Coder).

Task Type	**Polynomial-Time Tasks**						**NP-complete Tasks**					
Graph Scale	Small Graphs			Large Graphs			Small Graphs			Large Graphs		
LLM	Acc.	Fea.	Hallu.	Acc.	Fea.	Hallu.	Acc.	Fea.	Hallu.	Acc.	Fea.	Hallu.
Llama3-8b	28.6	45.3	53.9	20.2	58.1	41.7	9.4	19.7	78.2	1.4	31.2	68.8
Llama3-8b-SFT	**54.7**	**72.5**	**17.3**	**38.1**	**65.0**	**31.1**	**38.1**	**65.0**	**31.1**	**7.5**	**25.5**	**49.6**
GPT-4o	**71.9**	**82.3**	**17.5**	40.7	59.0	39.6	40.8	74.2	23.2	5.9	**49.2**	45.6
GPT-4o-Coder	62.7	71.7	18.9	**51.1**	**62.0**	**23.0**	**46.5**	**77.4**	**16.8**	**9.5**	46.7	**35.4**
DeepSeek-V2	51.4	64.7	35.0	24.7	38.0	61.2	**33.7**	**74.2**	24.5	**5.4**	**45.0**	55.0
DeepSeek-V2-Coder	**60.9**	**69.7**	**16.7**	**38.0**	**44.6**	**31.4**	31.2	63.1	**24.4**	4.0	23.4	**37.5**

architecture design. Beyond these comparisons, it's important to note that LLMs offer unique advantages beyond raw performance metrics. Their ability to interpret and respond to natural language problem descriptions significantly enhances accessibility for non-expert users, and their adaptability allows them to tackle new problem variations without explicit reprogramming.

3.2 In-depth Investigation into Hallucination

Several strategies have been proposed to mitigate hallucinations in LLMs [9, 38, 96, 83], but their effectiveness in GraphArena remains to be explored. We investigate three potential solutions:

Chain-of-Thought (CoT) Prompting. CoT prompting encourages models to generate intermediate reasoning steps towards a conclusion [84, 28, 58]. We designed manually crafted examples for the Graph Diameter and Connected Component tasks, evaluating the impact of zero to four CoT examples. Figure 5 demonstrates a general improvement in model accuracy. Deepseek-V2's performance significantly improves after adding one CoT example for the Component task. However, Claude3-haiku and Mixtral-7x8b show marginal improvements, and in some cases, performance degrades with more examples—a phenomenon also observed in previous benchmarks [81, 27]. These results suggest that while CoT prompting helps reduce errors, it remains insufficient to resolve hallucinations.

Instruction Tuning. LLMs often hallucinate on long-tail knowledge with limited training data [57, 42], and graph problem-solving likely falls into this category. To provide more extensive data exposure, we finetuned Llama3-8b on an additional 10,000 graph problems generated using the same process as GraphArena. This supervised mixed-task instruction tuning resulted in Llama3-8b-SFT. As shown in Table 2, Llama3-8b-SFT demonstrates substantial improvements over the base Llama3-8b model, with performance even comparable to Llama3-70b. Notably, hallucination rates on small graphs decreased significantly across all tasks. However, the improvement for large graphs was less pronounced, suggesting that long-context understanding remains a challenge.

Code Writing. LLMs often struggle with mathematical reasoning, which is crucial for solving complex graph problems. To address this, we explored using external tools for computation. Inspired by the approach of writing code for math problem-solving [30], we prompted LLMs (specifically GPT-4o and Deepseek-Coder [99]) to write and execute code for solving problems in GraphArena. Results in Table 2 show that this code-writing approach effectively reduced hallucination rates, particularly on large graphs. For instance, Deepseek-V2-Coder's hallucination rate for polynomial complexity tasks on large graphs decreased from 61.2% to 31.4%. However, we observed some negative impacts on small graphs, likely due to errors in extracting graph information from text and challenges in code writing.

4 Related Work

LLM Evaluation. Early evaluations of text generation focused on fundamental skills, including similarity measures such as ROUGE [51], METEOR [8], and BLEU [64] for machine translation [52, 69]. They also included inherent scores like Diversity [46] and Perplexity [37] for open-domain conversations [95, 48], and factual measures such as F-score and exact match for machine reading comprehension [67, 16] and question answering [100, 43]. Recently, numerous benchmarks have employed more advanced tests to assess the high-level capabilities of LLMs [34, 24, 55, 53, 17, 45, 80, 18, 76]. For instance, the Massive Multitask Language Understanding benchmark [33] covers 57 subjects across STEM, the humanities, the social sciences, and more. The Beyond the Imitation Game Benchmark [70] includes over 200 tasks, ranging from language understanding to interaction. Human preference remains a critical aspect of LLM evaluation [97, 47, 22]. Chatbot Arena [22] employs a pairwise comparison approach and leverages crowd-sourcing to ensure task diversity.

LLM on Graphs. The synergy between graphs and LLMs has sparked considerable interest due to their bi-directional benefits: integrating graph-based approaches can enhance the reasoning abilities of LLMs, enabling them to tackle complex logical tasks such as mathematical problem-solving [94], code generation [31], and knowledge reasoning [90, 63], among others. Conversely, LLMs offer powerful capabilities to augment graph analytics and learning, particularly for predictive tasks in text-attributed networks [27, 81, 20, 21, 75]. Compared with machine learning tasks on graphs such as node classification and anomaly detection [73, 74], graph computational problems pose more challenges for LLMs as they require a deeper understanding of structural information and long-range multi-step reasoning [56, 82, 72, 14]. GraphQA [27] and NLGraph [81] are two early benchmarks in this domain, focusing on basic graph problems using small-scale synthetic graphs. VisionGraph [50] and GITA [85] introduce multimodal graph reasoning tasks that require LLMs to reason over both text and image modalities. GraphWiz [19] introduces an instruction-tuning dataset designed to improve LLMs' graph reasoning capabilities.

5 Conclusion

This paper introduces GraphArena, a benchmarking tool designed to evaluate the reasoning capabilities of LLMs on graph computational challenges. GraphArena features a realistic graph collection, carefully selected tasks, and a rigorous evaluation framework. Our assessment of ten LLMs across 10,000 graph computational problems reveals persistent hallucination issues, particularly with larger graphs, complex tasks, and models with fewer parameters. We investigated three potential solutions to mitigate model hallucination: chain-of-thought prompting, instruction tuning, and code writing, each demonstrating unique strengths and limitations. Our findings suggest promising directions for future research: exploring advanced finetuning techniques to enhance LLMs' code writing and mathematical reasoning abilities, and developing better integration of external tools to support LLMs in graph problem-solving.

References

[1] Gemma 7b it. https://huggingface.co/google/gemma-7b-it. Accessed: 2024-05-25.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] AI@Meta. Llama 3 model card. *Github*, 2024.

[4] Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-yeol Ahn. Can we trust the evaluation on ChatGPT? In Anaelia Ovalle, Kai-Wei Chang, Ninareh Mehrabi, Yada Pruksachatkun, Aram Galystan, Jwala Dhamala, Apurv Verma, Trista Cao, Anoop Kumar, and Rahul Gupta, editors, *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 47–54, Toronto, Canada, July 2023. Association for Computational Linguistics.

[5] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[6] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[7] AI/ML API. https://aimlapi.com/. Accessed: 2024-05-25.

[8] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[9] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023, 2023.

[10] Reuven Bar-Yehuda and Shimon Even. A local-ratio theorem for approximating the weighted vertex cover problem. In *North-Holland Mathematics Studies*, volume 109, pages 27–45. Elsevier, 1985.

[11] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[12] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165, 2009.

[13] Ravi Boppana and Magnús M Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[14] Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*, 2023.

[15] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[16] Danqi Chen. *Neural reading comprehension and beyond*. Stanford University, 2018.

[17] Emma Chen, Aman Kansal, Julie Chen, Boyang Tom Jin, Julia Reisler, David E Kim, and Pranav Rajpurkar. Multimodal clinical benchmark for emergency care (mc-bec): A comprehensive benchmark for evaluating foundation models in emergency medicine. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 45794–45811. Curran Associates, Inc., 2023.

[18] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*, 2021.

[19] Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. Graphwiz: An instruction-following language model for graph problems. *arXiv preprint arXiv:2402.16029*, 2024.

[20] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024.

[21] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (LLMs). In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

[23] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. In *Operations Research Forum*, volume 3, page 20. Springer, 2022.

[24] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[25] Aliyun Dashscope. https://dashscope.aliyun.com/. Accessed: 2024-05-25.

[26] P ERDdS and A R&wi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

[27] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023.

[28] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024.

[30] Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations*, 2024.

[31] Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie LIU, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, Michele Tufano, Shao Kun Deng, Colin Clement, Dawn Drain, Neel Sundaresan, Jian Yin, Daxin Jiang, and Ming Zhou. Graphcode{bert}: Pre-training code representations with data flow. In *International Conference on Learning Representations*, 2021.

[32] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[33] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[34] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[35] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

[36] Yinya Huang, Xiaohan Lin, Zhengying Liu, Qingxing Cao, Huajian Xin, Haiming Wang, Zhenguo Li, Linqi Song, and Xiaodan Liang. Mustard: Mastering uniform synthesis of theorem and proof data. *arXiv preprint arXiv:2402.08957*, 2024.

[37] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[38] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023.

[39] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[40] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*, 2023.

[41] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[42] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR, 23–29 Jul 2023.

[43] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[44] Michael Ley. The dblp computer science bibliography: Evolution, research issues, perspectives. In *International symposium on string processing and information retrieval*, pages 1–10. Springer, 2002.

[45] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28541–28564. Curran Associates, Inc., 2023.

[46] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics, 2016.

[47] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[48] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In Greg Kondrak and Taro Watanabe, editors, *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[49] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.

[50] Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. *arXiv preprint arXiv:2405.04950*, 2024.

[51] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[52] Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[53] Junling Liu, Peilin Zhou, Yining Hua, Dading Chong, Zhongyu Tian, Andrew Liu, Helin Wang, Chenyu You, Zhenhua Guo, LEI ZHU, and Michael Lingzhi Li. Benchmarking large language models on cmexam - a comprehensive chinese medical exam dataset. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52430–52452. Curran Associates, Inc., 2023.

[54] Alejandro Lopez-Lira and Yuehua Tang. Can chatgpt forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*, 2023.

[55] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024.

[56] Zihan Luo, Xiran Song, Hong Huang, Jianxun Lian, Chenhao Zhang, Jinqi Jiang, Xing Xie, and Hai Jin. Graphinstruct: Empowering large language models with graph understanding and reasoning capability. *arXiv preprint arXiv:2403.04483*, 2024.

[57] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.

[58] William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2023.

[59] Maho Nakata and Tomomi Shimazaki. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.

[60] OpenAI. GPT-4 technical report. *Arxiv*, 2023.

[61] OpenFlights. https://openflights.org/. Accessed: 2024-05-25.

[62] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[63] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[64] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002.

[65] Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862*, 2024.

[66] DeepSeek Open Platform. https://www.deepseek.com/en. Accessed: 2024-05-25.

[67] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[68] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[69] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT 16. In Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi, editors, *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany, August 2016. Association for Computational Linguistics.

[70] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[71] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654, 2024.

[72] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

[73] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, pages 29628–29653, 2023.

[74] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 2022.

[75] Jianheng Tang, Kangfei Zhao, and Jia Li. A fused Gromov-Wasserstein framework for unsupervised knowledge graph entity alignment. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3320–3334, Toronto, Canada, July 2023. Association for Computational Linguistics.

[76] Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. Target-guided open-domain conversation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, Florence, Italy, July 2019. Association for Computational Linguistics.

[77] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008.

[78] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[79] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2017.

[80] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[81] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[82] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. *arXiv preprint arXiv:2402.08785*, 2024.

[83] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[84] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[85] Yanbin Wei, Shuai Fu, Weisen Jiang, Zejian Zhang, Zhixiong Zeng, Qi Wu, James T. Kwok, and Yu Zhang. Gita: Graph to visual and textual integration for vision-language graph reasoning, 2024.

[86] Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. *arXiv preprint arXiv:2405.14333*, 2024.

[87] Kexuan Xin, Zequn Sun, Wen Hua, Wei Hu, Jianfeng Qu, and Xiaofang Zhou. Large-scale entity alignment via knowledge graph merging, partitioning and embedding. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2240–2249, 2022.

[88] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

[89] Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*, 2024.

[90] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[91] Ruibin Yuan, Yinghao Ma, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, zhuo le, Yiqi Liu, Jiawen Huang, Zeyue Tian, Binyue Deng, Ningzhi Wang, Chenghua Lin, Emmanouil Benetos, Anton Ragni, Norbert Gyenge, Roger Dannenberg, Wenhu Chen, Gus Xia, Wei Xue, Si Liu, Shi Wang, Ruibo Liu, Yike Guo, and Jie Fu. Marble: Music audio representation benchmark for universal evaluation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 39626–39647. Curran Associates, Inc., 2023.

[92] Zhiping Zeng, Anthony KH Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1):25–36, 2009.

[93] Bohang Zhang, Jingchu Gai, Yiheng Du, Qiwei Ye, Di He, and Liwei Wang. Beyond weisfeiler-lehman: A quantitative framework for GNN expressiveness. In *The Twelfth International Conference on Learning Representations*, 2024.

[94] Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. Graph-to-tree learning for solving math word problems. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3928–3937, Online, July 2020. Association for Computational Linguistics.

[95] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[96] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[97] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023.

[98] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

[99] Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

[100] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50117–50143. Curran Associates, Inc., 2023.

A Additional Task and Dataset Information

We complete the description of the four remaining NP-complete tasks not fully detailed in the main text:

- **Minimum Vertex Cover (MVC):** Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, find a subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ such that each edge in $\mathcal{G}$ has at least one endpoint in $\mathcal{S}$. The Minimum Vertex Cover Problem is to (1) find the vertex cover $\mathcal{S}$ and (2) ensure the size of $\mathcal{S}$ is minimum among all vertex covers of $\mathcal{G}$. We explore this using the Social Network dataset.

- **Maximum Common Subgraph (MCS):** Compare two graphs $\mathcal{G}$ and $\mathcal{H}$ to find the largest common subgraph. The objectives are to (1) determine a node-induced common subgraph $\mathcal{S}$ between $\mathcal{G}$ and $\mathcal{H}$, and (2) maximize the size of $\mathcal{S}$. The PubChemQC dataset is used for this task.
- **Graph Edit Distance (GED):** For two graphs $\mathcal{G}$ and $\mathcal{H}$, determine the minimum edit distance via node mappings. This involves (1) establishing a node mapping that aligns $\mathcal{G}$ with $\mathcal{H}$, and (2) minimizing the edit operations required. These operations include adding or deleting an edge or isolated node, or relabeling a node. The PubChemQC dataset is utilized here.
- **Traveling Salesman Problem (TSP):** Solve the TSP in a weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $w : \mathcal{E} \to \mathbb{R}^+$ assigns a positive weight to each edge, representing the travel distance. The objectives are to (1) find a route $\mathcal{P}$ that visits each node exactly once and returns to the starting point, and (2) minimize the total travel distance. We utilize the OpenFlights dataset for this challenge. To ensure every pair of nodes is connected, thereby guaranteeing a feasible solution, we convert the dataset into a complete graph by adding edges to represent the shortest possible distances between nodes that are not directly connected.

Table 3 presents comprehensive statistics for each task in GraphArena, including the average and maximum number of nodes, edges, and text length (in characters). Tables 6-15 provide detailed examples of prompts for each task. For the Connected Component and Graph Diameter tasks, chain-of-thought prompting with step-by-step solution demonstrations is provided.

B Additional Experimental Results

Figure 6 offers a feasibility and accuracy comparison of the remaining five LLMs on individual tasks, complementing the analysis in Figure 2. Figures 7 and 8 illustrate the influence of graph size on hallucination probability for the remaining seven tasks, complementing Figure 3. Table 4 extends the analysis in Figure 4, showing the percentage of problems where each of the 10 LLMs wins, ties, or loses against Random, Greedy, and Approximated graph algorithms.

Comparison with graph neural networks (GNNs). Directly comparing GNNs and LLMs in GraphArena is challenging due to their fundamentally different paradigms. LLMs are multi-task, unsupervised models that process natural language inputs, while GNNs are typically task-specific, supervised models that operate on network data. We compared three representative GNNs: Graph Isomorphism Network (GIN [88]), Graph Attention Network (GAT [79]), and Graph Sample and Aggregate (GSAGE [32]). For simplicity, we configured GNNs to directly predict the answer rather than generating the solution path or component. Although it is feasible to let GNNs generate detailed solutions, different tasks like GED and MCS would require specific architectural modifications, adding unnecessary complexity. We generated an equivalent amount of training data for GNNs using the same framework as GraphArena. Table 5 shows the average accuracy for these GNNs on Polynomial-time and NP-complete problems across small and large graphs. The results indicate that GPT-4o demonstrates superior performance in most scenarios. While GNNs achieve the best results on large-scale NP-complete problems, recent studies have theoretically proved the limitations in GNNs' computational capabilities [93]. For instance, message passing neural networks cannot even count triangles. Given the limitations, we posit that GNNs might be performing pattern-based regression rather than genuinely solving graph problems. In contrast, LLMs appear more aligned with true problem-solving.

Table 3: Statistics of problems in each task of GraphArena, including the average and maximum number of nodes (V), edges (E), and text length (T).

Problems	*avg.* $\lvert V\rvert$	*max.* $\lvert V\rvert$	*avg.* $\lvert E\rvert$	*max.* $\lvert E\rvert$	*avg.* $\lvert T\rvert$	*max.* $\lvert T\rvert$
Common Neighbor	19.4	50	68.6	409	4323.5	18014
Shortest Distance	19.6	50	36.8	209	5639.5	23731
Connected Component	13.5	30	14.4	122	1785.4	5503
Graph Diameter	13.4	30	22.0	128	3741.3	11783
Maximum Clique Problem	13.2	30	39.1	154	3025.0	7921
Maximum Independent Set	12.8	30	19.5	141	1902.8	6182
Minimum Vertex Cover	14.1	30	23.3	151	2178.4	6429
Maximum Independent Set	9.2	20	8.9	22	1025.1	1187
Graph Edit Distance	9.4	20	9.2	23	1643.7	2142
Travelling Salesman Problem	9.5	20	50.6	190	1806.1	4741

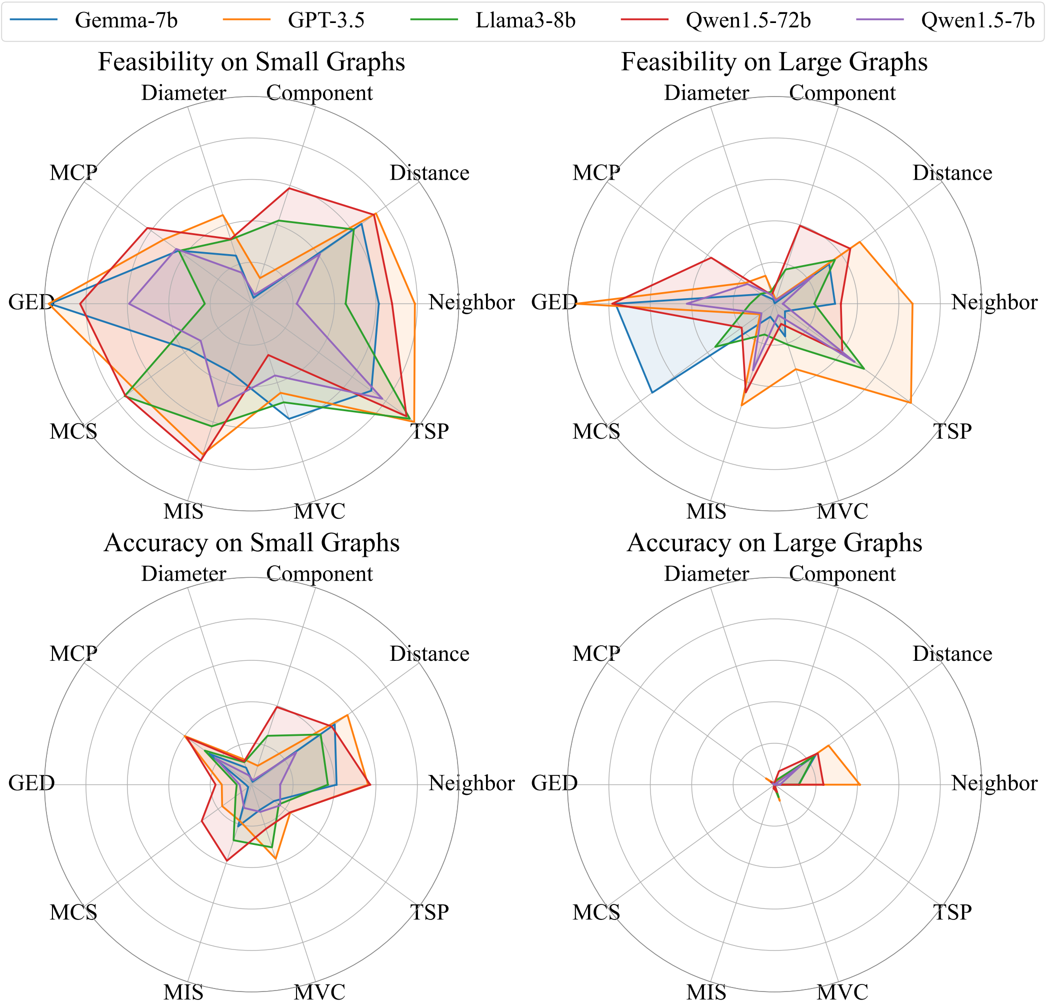


Figure 6: Feasibility and accuracy comparison of the remaining five LLMs in Figure 2 on each individual task.

Table 4: The percentage of problems where **LLM wins, ties, or loses against three types of classic graph algorithms**: Random, Greedy, and Approximated.

	Travelling Salesman Problem (Small)			**Travelling Salesman Problem (Large)**		
Algorithm	Random	Greedy	Approximated	Random	Greedy	Approximated
LLM	win/tie/lose	win/tie/lose	win/tie/lose	win/tie/lose	win/tie/lose	win/tie/lose
GPT-4o	67.4/16.2/16.4	20.6/43.4/36.0	15.4/33.0/51.6	**82.6**/0.0/17.4	**9.2**/2.8/88.0	**5.4**/0.4/94.2
GPT-3.5	58.4/13.6/28.0	13.0/30.2/56.8	7.4/24.0/68.6	66.8/0.0/33.2	0.4/0.0/99.6	0.2/0.0/99.8
Claude3-haiku	54.8/14.4/30.8	15.4/25.2/59.4	9.4/21.0/69.6	72.6/0.0/27.4	1.0/0.0/99.0	0.6/0.0/99.4
Llama3-70b	59.8/18.4/21.8	12.0/36.6/51.4	8.4/23.8/67.8	79.6/0.0/20.4	1.4/0.0/98.6	0.6/0.0/99.4
Llama3-8b	43.0/13.6/43.4	12.2/15.4/72.4	6.4/15.6/78.0	29.8/0.0/70.2	0.2/0.0/99.8	0.0/0.0/100.0
Qwen1.5-72b	49.4/16.0/34.6	13.2/25.0/61.8	7.0/22.8/70.2	26.4/0.0/73.6	0.0/0.0/100.0	0.0/0.0/100.0
Qwen1.5-7b	33.6/11.2/55.2	10.8/14.2/75.0	6.0/14.8/79.2	23.0/0.0/77.0	0.0/0.0/100.0	0.0/0.0/100.0
Deepseek-V2	**71.0**/15.4/13.6	**21.0**/42.0/37.0	**15.8**/32.2/52.0	67.2/0.0/32.8	4.4/0.2/95.4	2.2/0.0/97.8
Mixtral-7x8b	43.6/16.8/39.6	10.6/22.6/66.8	6.0/18.0/76.0	23.6/0.0/76.4	0.0/0.0/100.0	0.0/0.0/100.0
Gemma-7b	27.6/15.0/57.4	8.8/13.8/77.4	5.0/13.0/82.0	3.8/0.0/96.2	0.0/0.0/100.0	0.0/0.0/100.0
	Graph Edit Distance (Small)			**Graph Edit Distance (Large)**		
GPT-4o	63.4/14.0/22.6	31.0/41.8/27.2	10.0/25.8/64.2	77.4/1.2/21.4	34.8/14.4/50.8	25.8/2.2/72.0
GPT-3.5	62.6/16.6/20.8	35.2/27.0/37.8	6.4/18.4/75.2	93.0/1.0/6.0	51.2/9.8/39.0	33.4/4.8/61.8
Claude3-haiku	62.4/17.6/20.0	32.8/33.6/33.6	9.8/21.8/68.4	**93.8**/0.6/5.6	**58.0**/11.0/31.0	**39.2**/5.8/55.0
Llama3-70b	**67.6**/16.4/16.0	**41.2**/30.0/28.8	**10.6**/32.0/57.4	80.4/0.4/19.2	41.8/10.6/47.6	28.4/3.4/68.2
Llama3-8b	13.8/4.6/81.6	10.8/6.2/83.0	3.0/7.2/89.8	11.8/0.0/88.2	10.8/0.4/88.8	6.6/1.0/92.4
Qwen1.5-72b	52.2/15.6/32.2	28.6/21.8/49.6	5.4/18.4/76.2	72.6/1.4/26.0	27.6/7.4/65.0	18.8/3.0/78.2
Qwen1.5-7b	29.0/14.8/56.2	15.2/14.4/70.4	2.8/5.8/91.4	40.6/0.0/59.4	23.0/3.0/74.0	15.2/2.6/82.2
Deepseek-V2	63.0/13.6/23.4	35.4/32.8/31.8	10.6/26.8/62.6	82.4/1.0/16.6	45.6/13.2/41.2	32.6/4.4/63.0
Mixtral-7x8b	37.2/14.0/48.8	17.6/21.4/61.0	6.0/11.4/82.6	45.0/0.6/54.4	15.8/4.4/79.8	9.4/1.2/89.4
Gemma-7b	55.6/20.8/23.6	30.8/21.0/48.2	3.6/6.4/90.0	74.8/1.0/24.2	43.0/7.2/49.8	25.0/3.4/71.6
	Maximum Independent Set (Small)			**Maximum Independent Set (Large)**		
GPT-4o	59.4/13.8/26.8	0.2/51.6/48.2	1.2/53.2/45.6	33.4/1.2/65.4	0.0/3.8/96.2	0.2/5.2/94.6
GPT-3.5	46.8/18.6/34.6	0.0/18.6/81.4	0.2/19.2/80.6	41.2/2.4/56.4	0.0/0.8/99.2	0.0/1.0/99.0
Claude3-haiku	**61.6**/12.0/26.4	0.0/45.4/54.6	0.2/48.6/51.2	40.0/3.2/56.8	0.0/1.6/98.4	0.2/2.0/97.8
Llama3-70b	59.6/23.2/17.2	0.0/36.8/63.2	0.6/38.2/61.2	**43.8**/2.6/53.6	**0.2**/2.6/97.2	**0.6**/3.8/95.6
Llama3-8b	44.2/11.8/44.0	**0.4**/28.0/71.6	**1.4**/29.2/69.4	13.2/1.6/85.2	0.0/1.0/99.0	0.2/1.4/98.4
Qwen1.5-72b	55.6/16.0/28.4	0.0/39.0/61.0	0.8/39.8/59.4	33.6/3.8/62.6	0.0/1.8/98.2	0.0/2.2/97.8
Qwen1.5-7b	31.0/11.0/58.0	0.0/11.8/88.2	0.0/12.4/87.6	19.6/3.4/77.0	0.0/0.0/100.0	0.2/0.0/99.8
Deepseek-V2	53.2/15.8/31.0	0.0/36.2/63.8	0.8/37.4/61.8	12.6/0.6/86.8	0.0/3.2/96.8	0.4/3.6/96.0
Mixtral-7x8b	44.0/16.4/39.6	0.0/11.6/88.4	0.4/12.6/87.0	27.8/3.4/68.8	**0.2**/0.8/99.0	0.6/0.6/98.8
Gemma-7b	27.6/4.6/67.8	0.0/21.2/78.8	0.0/21.2/78.8	5.8/0.4/93.8	0.0/0.6/99.4	0.0/0.6/99.4
	Minimum Vertex Cover (Small)			**Minimum Vertex Cover (Large)**		
GPT-4o	37.8/6.6/55.6	32.6/8.8/58.6	26.4/15.0/58.6	24.0/1.0/75.0	**13.6**/4.6/81.8	**12.2**/4.2/83.6
GPT-3.5	39.8/5.2/55.0	36.8/6.8/56.4	28.8/14.8/56.4	24.0/1.6/74.4	10.4/5.8/83.8	10.2/5.8/84.0
Claude3-haiku	43.2/9.2/47.6	32.0/17.0/51.0	29.6/17.0/53.4	24.4/0.6/75.0	12.8/7.0/80.2	**12.2**/6.6/81.2
Llama3-70b	41.0/6.2/52.8	**40.0**/7.4/52.6	**32.0**/14.4/53.6	16.8/0.2/83.0	12.2/4.4/83.4	12.0/3.8/84.2
Llama3-8b	39.6/7.8/52.6	32.4/11.8/55.8	27.0/15.8/57.2	17.8/0.2/82.0	10.2/2.0/87.8	9.4/4.2/86.4
Qwen1.5-72b	22.2/2.4/75.4	19.8/5.4/74.8	17.0/7.6/75.4	7.8/0.6/91.6	4.4/1.6/94.0	4.8/1.4/93.8
Qwen1.5-7b	25.0/9.4/65.6	13.0/18.2/68.8	13.0/15.0/72.0	5.4/0.2/94.4	1.6/1.4/97.0	1.2/1.8/97.0
Deepseek-V2	**47.2**/8.6/44.2	36.6/14.8/48.6	31.6/17.0/51.4	**28.2**/2.0/69.8	10.6/8.4/81.0	11.2/8.8/80.0
Mixtral-7x8b	29.6/9.8/60.6	15.4/20.4/64.2	12.6/19.0/68.4	19.6/1.2/79.2	6.2/3.8/90.0	5.6/4.4/90.0
Gemma-7b	38.0/14.8/47.2	15.0/30.4/54.6	15.0/25.6/59.4	15.4/0.2/84.4	4.8/2.4/92.8	3.6/4.4/92.0

Table 5: Average accurac (%) for three representative GNNs on **P**olynomial-time and **NP**-complete problems across small and large graphs.

Models	P (Small)	P (Large)	NP (Small)	NP (Large)
GIN	59.9	39.6	33.8	**25.8**
GAT	42.3	22.5	32.3	17.9
GSAGE	41.0	22.6	33.7	17.6
GPT-4o	**71.9**	**40.7**	**40.8**	5.9

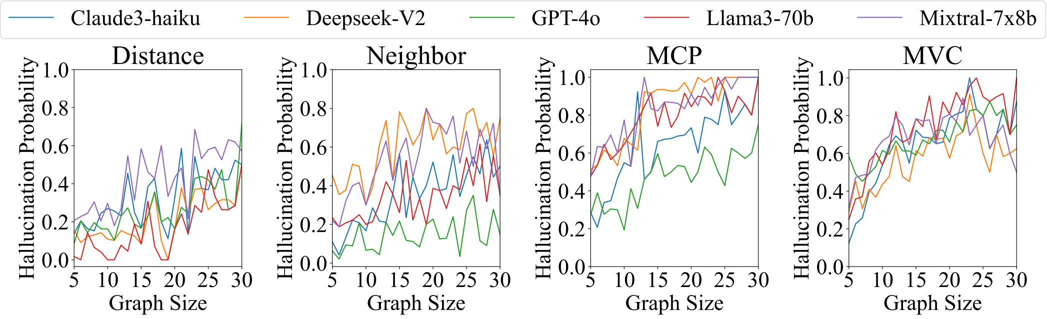


Figure 7: The influence of graph size on hallucination probability for the tasks of Distance, Neighbor, MCP, and MVC.

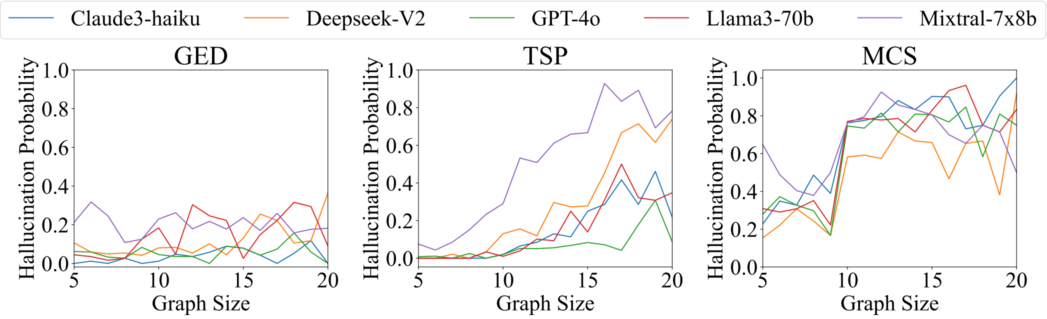


Figure 8: The influence of graph size on hallucination probability for the tasks of GED, TSP, and MCS.

Table 6: An example of CoT prompting with step-by-step demonstrations on the Component task.

CoT Prompts of the Connected Component Task. (1-shot)

You are required to identify all connected components in the given social network and output one representative node from each component.
Within a connected component, any node can be reached from any other node through the edges in the graph. Different connected components are isolated from each other.

Example

- Names in the network: Michelle Nelson, Kevin Brooks, Eric Davis, Toni Suarez, Teresa Hughes
- Friendship connections: Michelle Nelson to Teresa Hughes, Kevin Brooks to Eric Davis, Eric Davis to Toni Suarez

The answer includes one representative element from each connected component in the given social network: In this social network, we have the following names and connections: Michelle Nelson, Kevin Brooks, Eric Davis, Toni Suarez, and Teresa Hughes. The connections are: Michelle Nelson to Teresa Hughes, Kevin Brooks to Eric Davis, and Eric Davis to Toni Suarez. From these connections, we can identify two isolated connected components. The first component includes Michelle Nelson and Teresa Hughes, and the second component includes Kevin Brooks, Eric Davis, and Toni Suarez. By selecting one representative node from each connected component, we have Teresa Hughes from the first component and Kevin Brooks from the second component. Therefore, the answer is [Teresa Hughes, Kevin Brooks].

Problem to Solve

- Names in the network: Michelle Nelson, Kevin Brooks, Eric Davis, Toni Suarez, Teresa Hughes
- Friendship connections: Michelle Nelson to Teresa Hughes, Kevin Brooks to Eric Davis, Eric Davis to Toni Suarez

Identify all connected components in this network. Note that for each connected component, you should only output one of its nodes. Present your answer in the following format: [UserA, UserB, UserC, UserD, ...]

Table 7: An example of CoT prompting with step-by-step demonstrations on the Diameter task.

CoT Prompts of the Graph Diameter Task. (1-shot)

You are required to calculate the diameter of an undirected knowledge graph.
The diameter of a graph is the maximum distance between any pair of nodes in the graph. To compute this, you need to find the shortest path between all pairs of nodes and then determine the maximum length of these shortest paths.

Example

-Entities in this knowledge graph: SeineetMarne, MontereauFaultYonne, Vittorio De Sica, SaintFiacre SeineetMarne, France
- The relationships between these entities are as follows: SeineetMarne is connected to SaintFiacre SeineetMarne via the relationship department. SeineetMarne is connected to MontereauFaultYonne via the relationship department. SeineetMarne is connected to France via the relationship country. MontereauFaultYonne is connected to France via the relationship country. Vittorio De Sica is connected to France via the relationship deathPlace. SaintFiacre SeineetMarne is connected to France via the relationship country.

To calculate the diameter of this undirected knowledge graph, we first need to understand the connections and the structure of the graph. The entities and their relationships are as follows: SeineetMarne is connected to SaintFiacre SeineetMarne via the relationship ’department’, SeineetMarne is connected to MontereauFaultYonne via the relationship ’department’, SeineetMarne is connected to France via the relationship ’country’, MontereauFaultYonne is connected to France via the relationship ’country’, Vittorio De Sica is connected to France via the relationship ’deathPlace’, and SaintFiacre SeineetMarne is connected to France via the relationship ’country’. The graph structure can be visualized as follows: SeineetMarne is connected to SaintFiacre SeineetMarne, MontereauFaultYonne, and France. MontereauFaultYonne and SaintFiacre SeineetMarne are also connected to France, and France is connected to Vittorio De Sica. By analyzing the shortest paths between all pairs of nodes, we find that SeineetMarne to SaintFiacre SeineetMarne, SeineetMarne to MontereauFaultYonne, SeineetMarne to France, MontereauFaultYonne to France, SaintFiacre SeineetMarne to France, and France to Vittorio De Sica all have a direct connection with 1 edge. The longest shortest paths are from SeineetMarne, MontereauFaultYonne, and SaintFiacre SeineetMarne to Vittorio De Sica via France, each with 2 edges. Therefore, the diameter of this knowledge graph is 2, and one of the paths that represent this diameter is [SeineetMarne, France, Vittorio De Sica].

Problem to Solve

- Entities in this knowledge graph: PlayStation 2, Buzz! Junior: Jungle Party, Tekken (video game), Buzz!, PlayStation Network
- The relationships between these entities are as follows:
PlayStation 2 is connected to Tekken (video game) via the relationship computingPlatform. PlayStation 2 is connected to Buzz! Junior: Jungle Party via the relationship computingPlatform. Buzz! Junior: Jungle Party is connected to Buzz! via the relationship series. Buzz! Junior: Jungle Party is connected to PlayStation Network via the relationship computingPlatform. Tekken (video game) is connected to PlayStation Network via the relationship computingPlatform.

Please determine the diameter of this network and output the corresponding path in the following format: [Entity1, Entity2, ..., EntityN].

Table 8: An example of the Common Neighbor task.

Prompts of the Common Neighbor Task. (1-shot)

Your task is to find the common neighbors of two nodes in an undirected academic network. In this network, nodes represent authors and edges represent research collaborations.

Example

- Authors in the network: Marie-Francine Moens, Stefanie Brüninghaus, Henry Prakken, David W. Aha, Kevin D. Ashley, Ronald Prescott Loui
- Research collaborations between these authors: Marie-Francine Moens and Kevin D. Ashley, Marie-Francine Moens and Stefanie Brüninghaus, Stefanie Brüninghaus and David W. Aha, Stefanie Brüninghaus and Henry Prakken, Stefanie Brüninghaus and Kevin D. Ashley, Stefanie Brüninghaus and Ronald Prescott Loui, Henry Prakken and Kevin D. Ashley, Henry Prakken and Ronald Prescott Loui, David W. Aha and Kevin D. Ashley, Kevin D. Ashley and Ronald Prescott Loui.

Common neighbors between Marie-Francine Moens and Stefanie Brüninghaus: [Kevin D. Ashley]
Problem to Solve

- Authors in the network: Sang Wan Lee, Hyoyoung Jang, Z. Zenn Bien, Zeungnam Bien
- Research collaborations between these authors: Sang Wan Lee and Zeungnam Bien, Sang Wan Lee and Z. Zenn Bien, Sang Wan Lee and Hyoyoung Jang, Hyoyoung Jang and Zeungnam Bien, Z. Zenn Bien and Zeungnam Bien.

Please identify the common neighbors of Sang Wan Lee and Hyoyoung Jang in this network. Present your answer in the following format: [AuthorA, AuthorB, AuthorC, AuthorD, ...].

Table 9: An example of the Shortest Distance task.

Prompts of the Shortest Distance Task. (1-shot)

Your task is to identify the shortest path between two specified entities in an undirected knowledge graph, minimizing the number of hops.

Example

- Entities in this knowledge graph: Chordata, Animalia, Sloggett's vlei rat, Mammalia
- The relationships between these entities are as follows:

Chordata is connected to Sloggett's vlei rat via the relationship phylum. Animalia is connected to Sloggett's vlei rat via the relationship kingdom. Sloggett's vlei rat is connected to Mammalia via the relationship class.
One shortest path between Animalia and Sloggett's vlei rat is: [Animalia, Sloggett's vlei rat]

Problem to Solve

- Entities in this knowledge graph: United States, Post-hardcore, Head Wound City, Texas, Cody Votolato
- The relationships between these entities are as follows:

United States is connected to Texas via the relationship country. United States is connected to Cody Votolato via the relationship birthPlace. Post-hardcore is connected to Cody Votolato via the relationship genre. Head Wound City is connected to Cody Votolato via the relationship associatedMusicalArtist. Texas is connected to Cody Votolato via the relationship birthplace.

Please determine the shortest path between Texas and United States in this network.Submit your answer in the format: [Entity1, Entity2, ..., EntityN], where Entity1 and EntityN are the specified start and end entities, and Entity2 through EntityN-1 are the intermediate entities on the shortest path.

Table 10: An example of the Graph Edit Distance task.

Prompts of the GED Task. (1-shot)

You are required to solve the Graph Edit Distance problem between two molecules. Each edit operation (adding or deleting an edge, adding or deleting an isolated node, or relabeling a node) has an identity cost. Your objective is to establish a mapping between the atom IDs from Molecule A to Molecule B, ensuring that each atom ID in Molecule A corresponds to exactly one atom ID in Molecule B. The mapping corresponds to the minimum edit cost between the two graphs.

Example

Molecule A:
- Atoms: N (atom 0), O (atom 1), Si (atom 2), O (atom 3), O (atom 4).
- Bonds: 0-1, 1-2, 2-3, 2-4.

Molecule B:
- Atoms: F (atom 0), B (atom 1), C (atom 2), N (atom 3), Br (atom 4).
- Bonds: 0-1, 1-2, 1-4, 2-3.

One optimal node mapping: [3, 2, 1, 0, 4].

Problem to Solve

You are given the following two molecules:

Molecule A:
Atoms: N (atom 0), C (atom 1), N (atom 2), F (atom 3).
Bonds: 0-1, 1-2, 1-3.
Molecule B:
Atoms: O (atom 0), C (atom 1), F (atom 2), F (atom 3).
Bonds: 0-1, 1-2, 1-3.

Represent the node mapping as a list of integers, where the position in the list corresponds to the atom ID in Molecule A and the value at that position indicates the corresponding atom ID in Molecule B.

For instance, if atom 0 in Molecule A corresponds to atom 1 in Molecule B, atom 1 in Molecule A corresponds to atom 0 in Molecule B, and atom 2 remains unchanged, the mapping would be represented as [1, 0, 2, ...].

Table 11: An example of the Maximum Clique Problem.

Prompts of the MCP Task. (1-shot)

You are required to solve the Maximum Clique Problem for an undirected academic network. In this network, nodes represent authors and edges represent research collaborations. Your objective is to find the largest subset of nodes such that every pair of vertices in this subset is connected by an edge.

Example

- Authors in the network: Keng Peng Tee, Veit Hagenmeyer, Bartosz Käpernick, Karl Henrik Johansson, Darryl DeHaan, James B. Rawlings, Andreas Kugi, Knut Graichen, Tilman Utz, Christian Ebenbauer.
- Research collaborations between these authors: Keng Peng Tee and James B. Rawlings, Keng Peng Tee and Darryl DeHaan, Veit Hagenmeyer and Knut Graichen, Bartosz Käpernick and Andreas Kugi, Bartosz Käpernick and Knut Graichen, Karl Henrik Johansson and Christian Ebenbauer, Darryl DeHaan and Knut Graichen, Darryl DeHaan and Christian Ebenbauer, James B. Rawlings and Knut Graichen, James B. Rawlings and Christian Ebenbauer, Andreas Kugi and Knut Graichen, Knut Graichen and Tilman Utz.

One Maximum Clique: [Knut Graichen, Bartosz Käpernick, Andreas Kugi].

Problem to Solve

- Authors in the network: Manfred Schmidt-Schauss, David Sabel, Manfred Schmidt-Schauß, Guillem Godoy.
- Research collaborations between these authors: Manfred Schmidt-Schauss and David Sabel, Manfred Schmidt-Schauss and Manfred Schmidt-Schauß, Manfred Schmidt-Schauss and Guillem Godoy, David Sabel and Manfred Schmidt-Schauß, Manfred Schmidt-Schauß and Guillem Godoy.

Identify the clique with the maximum number of authors in this network. Present your answer in the following format: [AuthorA, AuthorB, AuthorC, AuthorD, ...].

Table 12: An example of the Maximum Common Subgraph Task.

Prompts of the MCS Task. (1-shot)

You are required to solve the Maximum Common Subgraph problem. Your goal is to identify the common subgraph with the maximum number of atoms shared between the two molecules.

Example

Molecule A consists of 8 atoms with the following 9 bonds: 0-1, 0-6, 1-2, 2-3, 3-4, 3-7, 4-5, 5-6, 5-7.
Molecule B consists of 7 atoms with the following 7 bonds: 0-1, 1-2, 1-4, 2-3, 3-4, 3-6, 4-5.
One max common subgraph: [2, 3, 4, 5, 7, 6], [0, 1, 2, 3, 4, 6].

Problem to Solve

You are given the following two molecules:
Molecule A consists of 4 atoms with the following 3 bonds: 0-1, 1-2, 2-3. Molecule B consists of 4 atoms with the following 3 bonds: 0-1, 1-2, 1-3.
Provide the indices of the atoms in the common subgraph for each molecule in the following format: [Node indices in molecular A], [Node indices in molecular B].

For example, if the common subgraph is the subgraph of atom 1, 2, 3 in molecule A and the subgrah of atom 2, 3, 4 in molecule B, you should answer: [1, 2, 3], [2, 3, 4].

Table 13: An example of the Maximum Independent Set task.

Prompts of the MIS Task. (1-shot)

Your task is to solve the Maximum Independent Set problem in the given social network. In this network, each node represents a user, and each edge represents a friendship connection. You need to identify the largest subset of users such that no two users in this subset are friends connected by an edge.

Example

- Users in the network: Melinda Vaughan, Mary Thornton, Jeremiah Griffith, Lisa Anderson, Alfred Powell.
- Fiendship connections: Melinda Vaughan and Jeremiah Griffith, Mary Thornton and Jeremiah Griffith, Jeremiah Griffith and Lisa Anderson, Jeremiah Griffith and Alfred Powell.

One Maximum Independent Set: [Melinda Vaughan, Lisa Anderson, Alfred Powell, Mary Thornton].

Problem to Solve

- Users in the network: William Lawson, Daniel Shelton, Michelle Lewis, Julie Hayes.
- Friendship connections: William Lawson and Daniel Shelton, William Lawson and Julie Hayes, Michelle Lewis and Julie Hayes.

Identify the Maximum Independent Set of this network and present your answer in the following format: [UserA, UserB, UserC, UserD, ...].

Table 14: An example of the Minimum Vertex Cover task.

Prompts of the MVC Task. (1-shot)

Your task is to solve the Minimum Vertex Cover problem in the given social network. In this network, each node represents a user, and each edge represents a friendship connection. You need to identify the smallest subset of users such that every friendship connection has at least one user from this subset.

Example

- Users in the network: Julie Harris, David Torres, Vanessa Parker, Shawn Barnett, Karl Dean.
- Fiendship connections: Julie Harris and Vanessa Parker, Julie Harris and David Torres, Julie Harris and Shawn Barnett, Julie Harris and Karl Dean, David Torres and Vanessa Parker, David Torres and Shawn Barnett, David Torres and Karl Dean, Vanessa Parker and Shawn Barnett, Shawn Barnett and Karl Dean.

One Minimum Vertex Cover: [Julie Harris, David Torres, Shawn Barnett].

Problem to Solve

- Users in the network: Pamela Haynes, Kyle Meadows, Adam Nichols, Anna Lowery, Heather Dixon, Matthew Lee, Elizabeth Wood, Stephen Hess.
- Friendship connections: Pamela Haynes and Stephen Hess, Kyle Meadows and Matthew Lee, Kyle Meadows and Stephen Hess, Kyle Meadows and Adam Nichols, Adam Nichols and Stephen Hess, Adam Nichols and Heather Dixon, Anna Lowery and Stephen Hess, Heather Dixon and Stephen Hess, Matthew Lee and Stephen Hess, Elizabeth Wood and Stephen Hess.

Identify the Minimum Vertex Cover of this network and present your answer in the following format: [UserA, UserB, UserC, UserD, ...].

Table 15: An example of the Travelling Salesman Problem.

Prompts of the TSP Task. (1-shot)

You are required to solve the Travelling Salesman Problem for an undirected flight route network. Your objective is to determine the shortest possible route that visits each of the listed airports exactly once and returns to the starting point.

Example

- Airports to visit: VPY, AAE, BGA, YWB.
- Travel distances (in kilometers) between each pair of airports:

VPY to YWB: 16285; VPY to AAE: 9488; VPY to BGA: 13255; AAE to YWB: 7807; AAE to BGA: 9332; BGA to YWB: 6575.
One shortest route: [VPY, AAE, YWB, BGA, VPY].

Problem to Solve

- Airports to visit: YNT, TEB, CFC, VIE, NSH, ROV.
- Travel distances (in kilometers) between each pair of airports:

YNT to VIE: 7962; YNT to ROV: 6644; YNT to NSH: 6176; YNT to CFC: 18668; YNT to TEB: 12263; TEB to VIE: 6987; TEB to ROV: 8529; TEB to NSH: 10389; TEB to CFC: 8846; CFC to VIE: 10716; CFC to ROV: 12244; CFC to NSH: 13193; VIE to ROV: 1736; VIE to NSH: 3406; NSH to ROV: 3000.

Please calculate the shortest tour and format your answer as follows: [Airport A, Airport B, Airport C, ..., Airport A]
Identify the Minimum Vertex Cover of this network and present your answer in the following format: [UserA, UserB, UserC, UserD, ...].