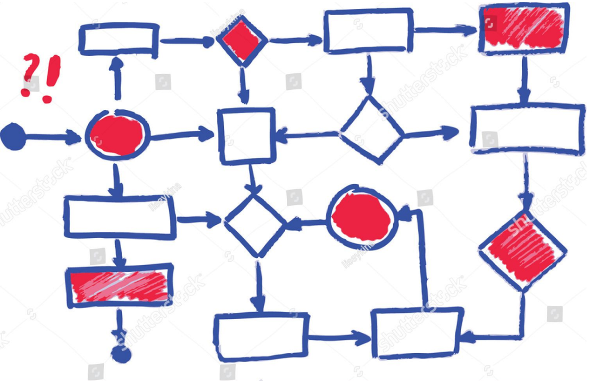
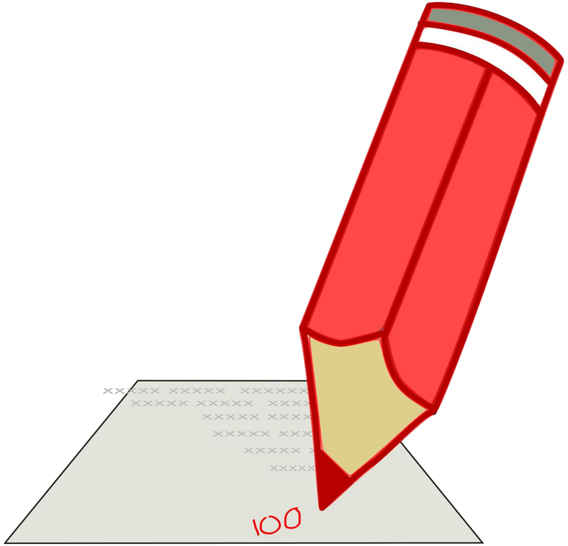
Enhanced Diagram Evaluation In Education System By Integrating Graph-Based Logic Analysis with LLM-Assisted Contextual Understanding





Abstract:

In **automated grading systems** for **education**, evaluating **diagrams**, such as **flowcharts**, often overlooks deeper aspects like **logic** and **problem-solving**. While **Large Language Models (LLMs)** provide **contextual evaluation**, they may miss **logical consistency**. This paper proposes an enhanced system combining **graph-based analysis** with **LLM-assisted evaluation**. By using **graph traversal algorithms**, we assess the **logical flow** and **structural correctness** of diagrams, while the LLM evaluates the **semantic meaning** of diagram text. This **hybrid approach** improves grading accuracy by addressing both **structural** and **contextual** aspects, ensuring better learning outcomes in academics.

Keywords:

Automated grading, education, **diagram evaluation**, **graph-based analysis**, **LLMs**, **logical flow**, structural correctness.

Introduction

1. **Automated Grading** : The use of technology for grading in academic subject is growing, particularly for diagram-heavy content like flowcharts.
2. **Challenges with Current Approaches**: Existing methods focus on structural and textual correctness but fail to fully assess the **logic** and **problem-solving** approaches in diagrams.
3. **Role of LLMs**: While **LLMs** provide **contextual understanding**, they often miss deeper **logical relationships** between diagram components.
4. **Need for Enhanced Evaluation**: A more comprehensive approach is needed to analyze not just text but also the **logical flow** and structure of diagrams.
5. **Objective**: This paper proposes a **hybrid system** combining **graph-based analysis** and **LLM evaluation** to improve the accuracy and depth of **automated grading** in education.

Model Answer

Answer 4. Heaps and stacks are both data structures used in computer science, but they serve different purposes and have different advantages. Here are some advantages of a heap over a stack: Dynamic Memory Allocation: Heaps allow for dynamic memory allocation, which means you can allocate memory as needed during program execution. This is useful for storing data structures that need to grow or shrink in size, such as arrays or linked lists. Variable Lifetime: Objects allocated on the heap have a variable lifetime, meaning they can exist beyond the scope of the function that created them. This allows you to create objects that persist even after the function has returned. Efficient Memory Management: Heaps use more sophisticated memory management techniques, such as memory pooling and garbage collection, which can lead to more efficient memory usage compared to stacks. Large Memory Allocation: Heaps can allocate larger amounts of memory than stacks, which are typically limited in size. Flexibility: Heaps offer more flexibility in terms of memory management, allowing you to allocate and deallocate memory in a more dynamic and controlled manner. Overall, the heap is useful for managing memory that needs to be dynamically allocated and deallocated, while the stack is more suitable for managing function call execution and local variable storage.

Input Answer

Answer 4 Advantages of heap over stack

Dynamic Memory Allocation - Heaps allow for dynamic memory allocation, which means you can that needs to grow or shrink in size.

Variable Lifetime

Objects allocated on the heap have a variable lifetime, meaning they can exist beyond the scope of the function that created them. This allows you to create objects that persists even after the function has returned

Efficient Memory Management

Heaps use more sophisticated memory management techniques, such as memory pooling and garbage collection, which can lead to more efficient memory usage compared to stacks

Large memory allocation

Heaps can allocate larger amounts of memory than stacks, which are typically limited in size

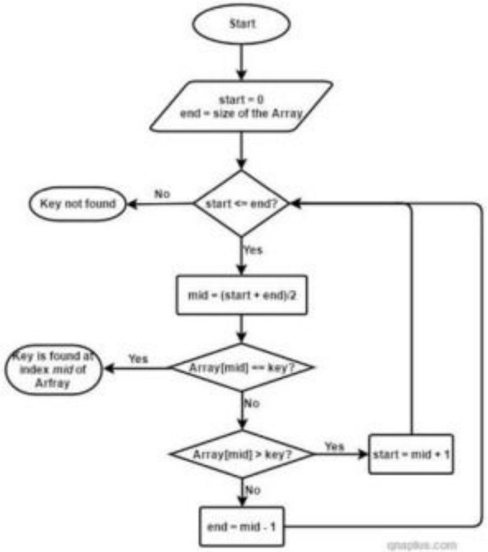
Flexibility

Heaps offer more flexibility in terms of memory management, allowing you to allocate and deallocate memory in a more dynamic and controlled manner

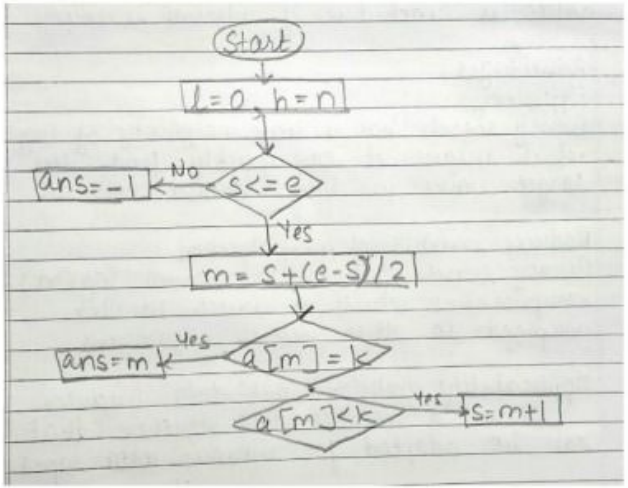
Result

3	**The input answer covers important points such as dynamic memory allocation, variable lifetime, efficient memory management, and flexibility. However, it is missing some details on the advantages of heaps over stacks, such as memory pooling and garbage collection. Additionally, the sentence structure could be improved for clarity and readability.**

Model Answer



Input Answer



Result

Score	Reason
0.875	The input flowchart has missing blocks, incorrect order of blocks, and incorrect connections between blocks. The text inside the blocks is also incorrect.

Real time tool

Aspect	Gap Identified
Evaluation Metrics	Lack of multi-dimensional evaluation for flowchart comprehension in MLLMs. Existing evaluations focus primarily on performance, neglecting reasoning and logical verification.
Model Performance	Even top-performing models like GPT-4 score suboptimally (56.63), indicating a gap in MLLM's ability to fully comprehend flowchart logic.
Benchmarking	Existing benchmarks for flowchart comprehension are limited, requiring new frameworks like *FlowCE* to address the gaps in current model evaluation.
Open-Source Models	Open-source models (e.g., Phi-3-Vision) lag behind closed-source models in comprehension tasks, highlighting a gap in accessible model capabilities.

Literature Review

Paper Name	Year	Author(s)	Drawback/Research Gap
[2407.10793] GraphEval: A Knowledge-Graph Based LLM Hallucination Evaluation Framework	2024	Hannah Sansford, Nicholas Richardson, et al.	Limited to analysing hallucinations and focusing on hallucination correction. Lacks practical use of graph based corrections.
[2409.15749] Automated Assessment of Multimodal Answer Sheets in the STEM domain (arxiv.org)	2024	Rajlaxmi Patil, et al.	Limited to using LLMs for evaluating diagrams, which may miss logical relationships and flow consistency.
Recognition of Handwritten Flowcharts using Convolutional Neural Networks (ijcaonline.org)	2022	C. David Betancourt Montellano, et al.	Focuses on handwritten flowchart recognition but lacks advanced logical evaluation of flow consistency.
Online recognition of sketched arrow-connected diagrams | International Journal on Document Analysis and Recognition (IJDAR) (springer.com)	2016	Martin Bresler, Daniel Pruša, et al.	Only recognizes diagrams; does not incorporate a logical evaluation of the process or decision flows.

Aspect	Existing Methodology	Proposed Methodology
Logical Flow Evaluation	Evaluates diagrams mostly for **textual correctness**, missing subtle **logical inconsistencies** or improper decision branches.	**Graph-Based Analysis** ensures proper **logical flow**, detecting missing connections, inconsistent decision branches, and feedback loops for a deeper assessment of diagram logic.
Error Detection	Limited to identifying **textual errors**, often missing structural or logical flaws in the overall process of the diagram.	Provides **advanced error detection**, identifying specific **logical mistakes** such as incorrect flows, missing links, or incomplete processes, offering **detailed feedback**.
Comprehe nsive Grading	Focuses on **structural correctness** and text, but often misses evaluating the diagram's **problem-solving approach** and logical integrity.	Combines **graph traversal** and **LLM understanding** to evaluate the **structure, logic, and problem-solving approach**, ensuring a more **comprehensive** and accurate assessment.

Market Research

Platforms like Coursera, edX, and Khan Academy use AI to grade and provide feedback on STEM-related assignments, including diagrammatic problems. However, these systems are often text-focused and lack advanced capabilities for logical or semantic diagram evaluation.

E-education platforms, their Generative AI chapter - The Hindu

Stanford has developed and tested AI models for automated grading as part of its AI and education research initiatives, emphasizing methods like OCR and semantic analysis.

Grade Like a Human: Rethinking Automated Assessment with Large Language Models | SCALE Initiative

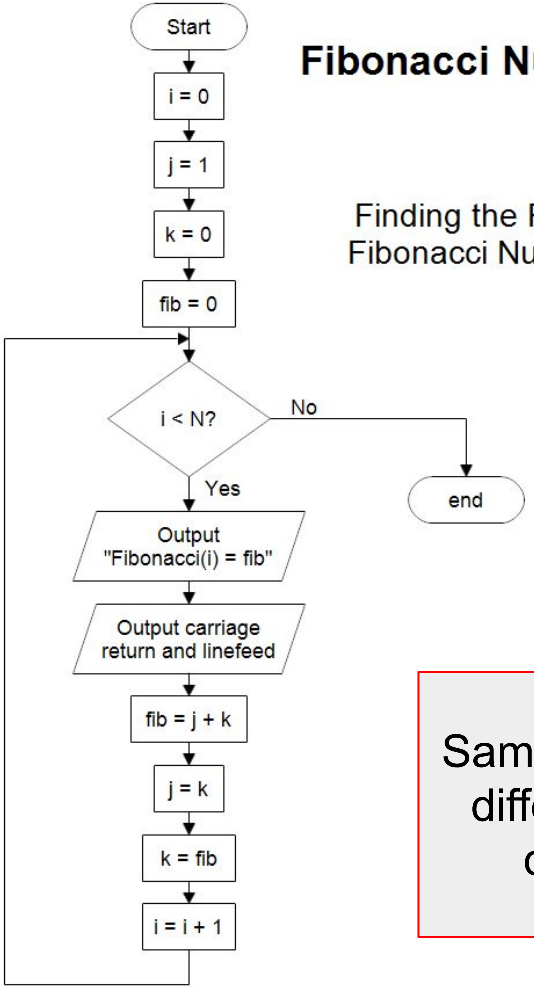
AI Grading in Peer Reviews: Enhancing Coursera’s Learning Experience with Faster, High-Quality Feedback - Coursera Blog

Problem Statement:

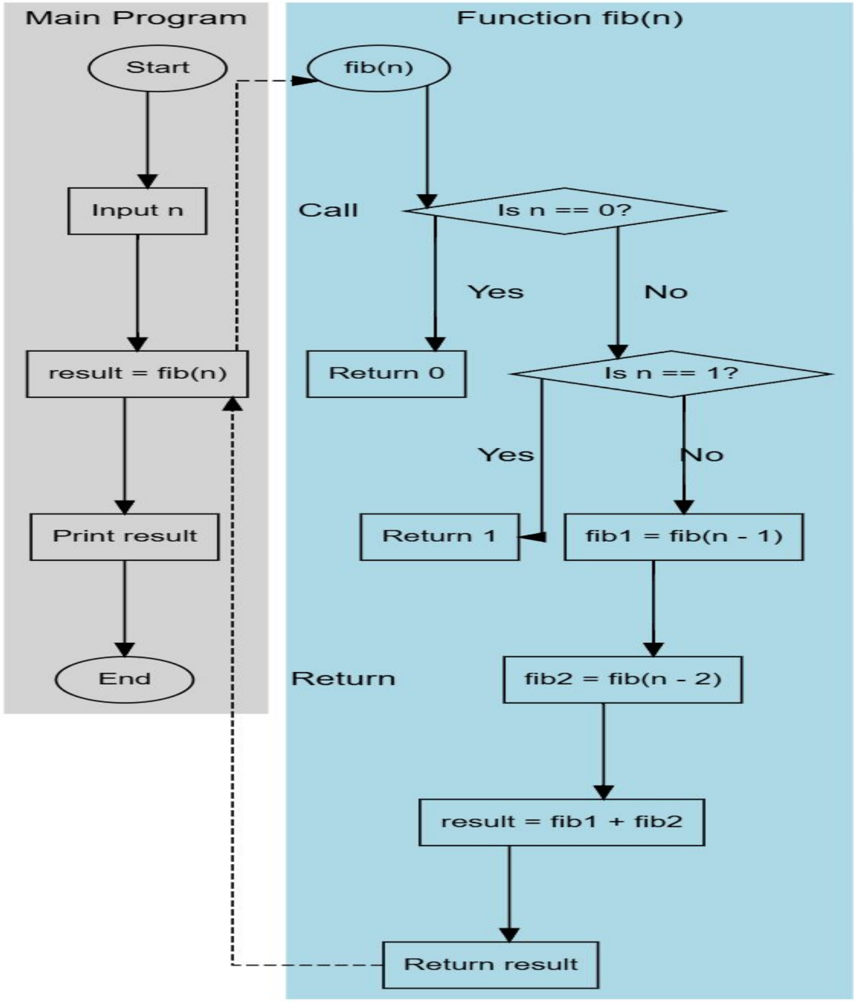
Traditional grading methods in an **education** system require significant teacher effort to evaluate the **structural correctness** and **logical integrity** of **diagrams** like flowcharts. Existing automated systems fall short by focusing solely on **textual evaluation**, necessitating a hybrid approach combining **graph-based analysis** with **LLM-based contextual evaluation** for comprehensive and efficient assessments.

Fibonacci Numbers

Finding the First N Fibonacci Numbers

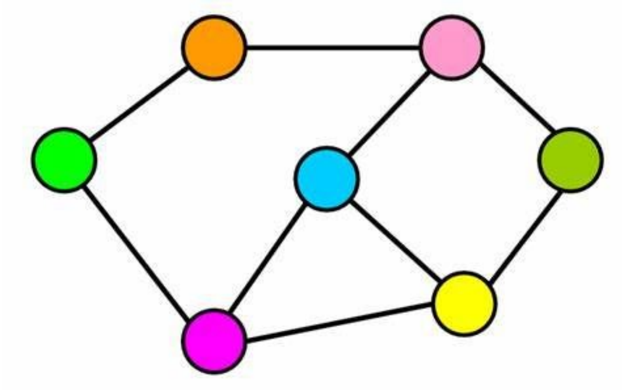


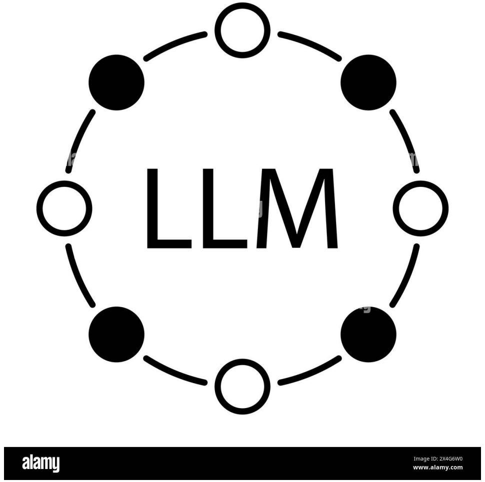
Same solution but different logical diagrams.



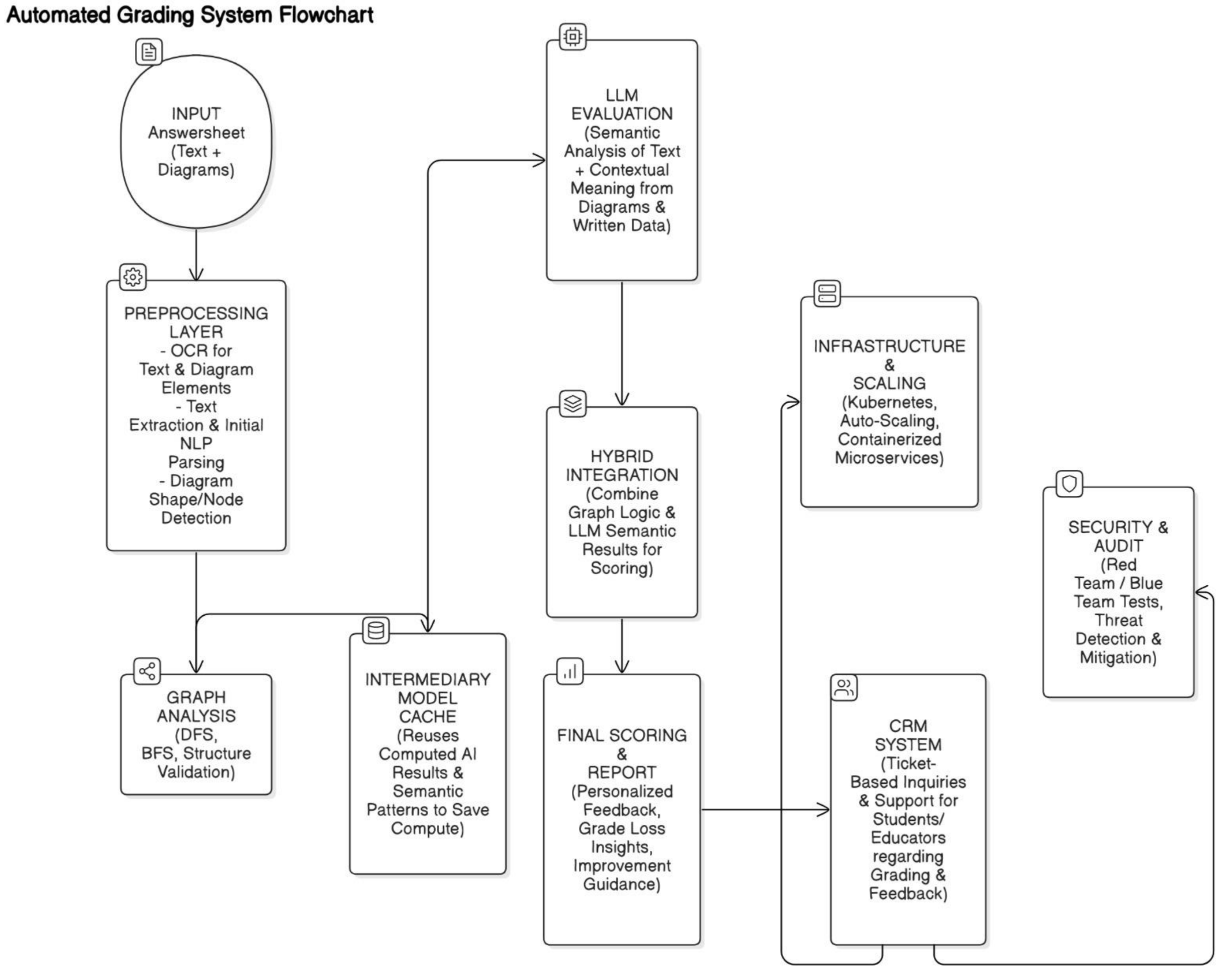
Proposed Solution:

To significantly enhance the evaluation of diagrams in education, we propose a **hybrid evaluation system** that integrates **graph-based analysis** with **LLM-based contextual evaluation.** This solution will focus on improving **logical flow assessment**, **error detection**, and overall **grading accuracy**.

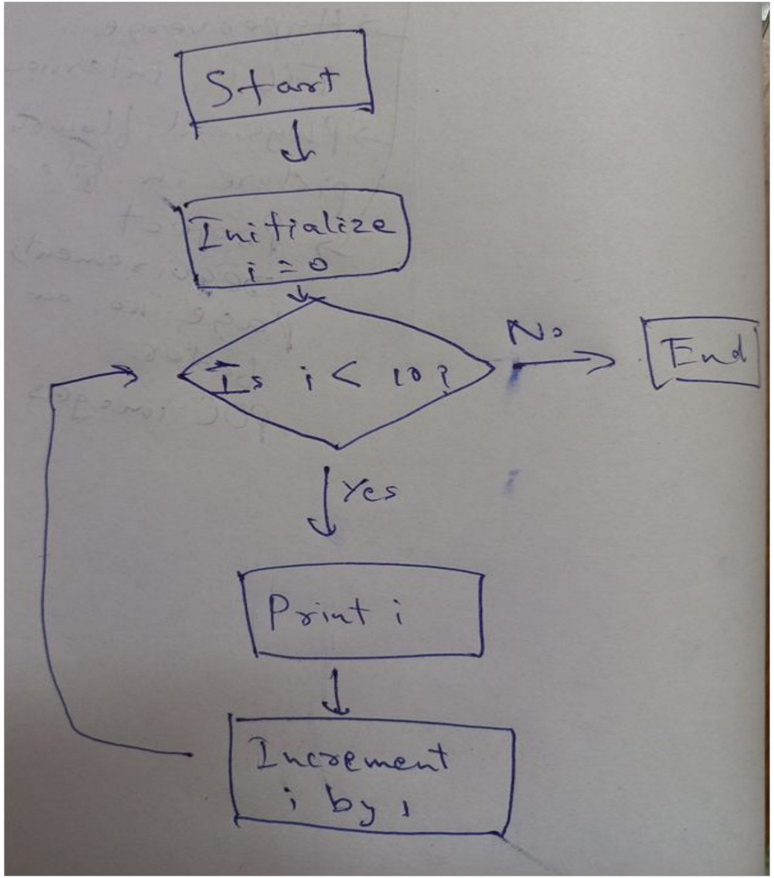




ARCHITECTURAL DIAGRAM



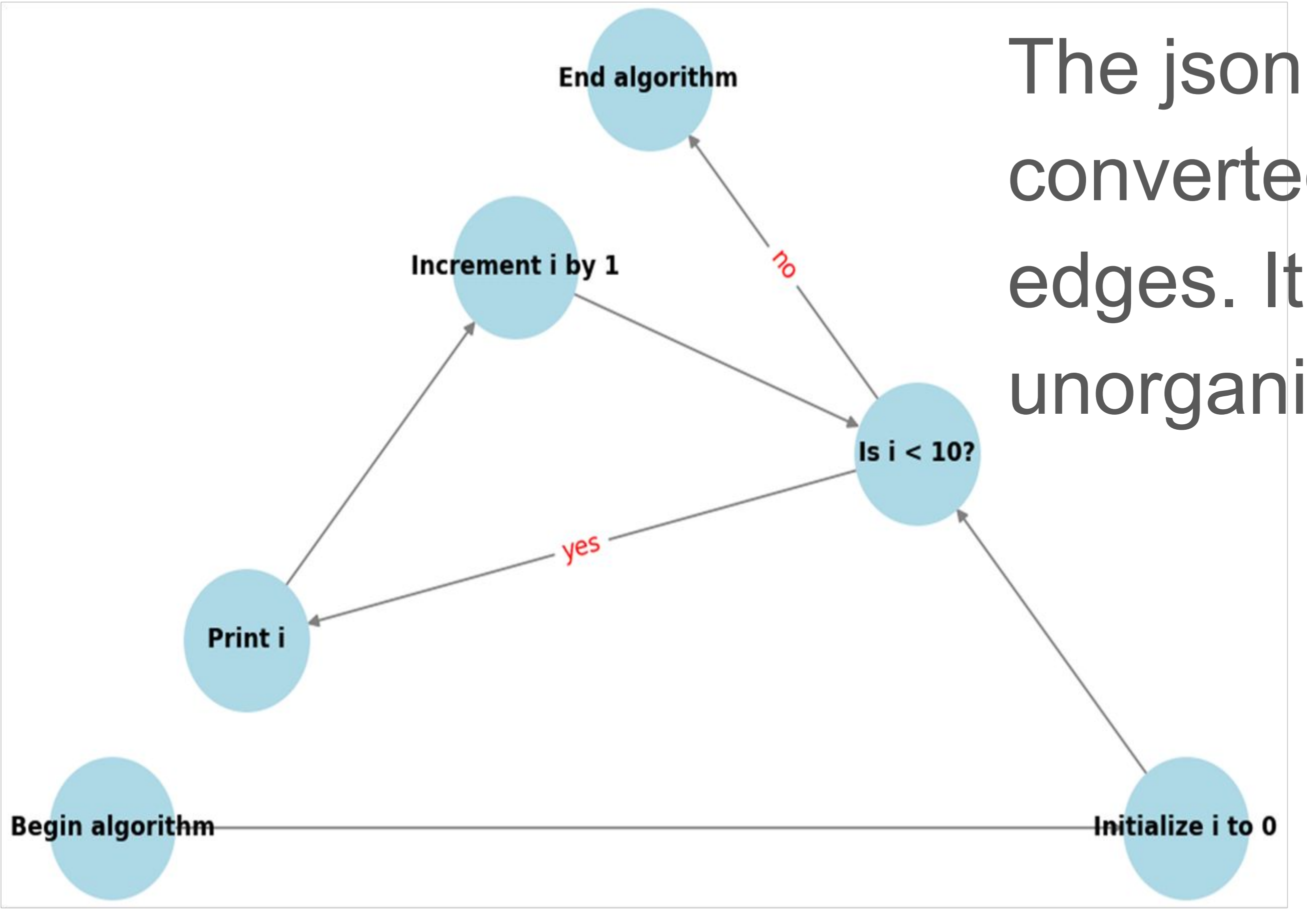
PROOF OF CONCEPT



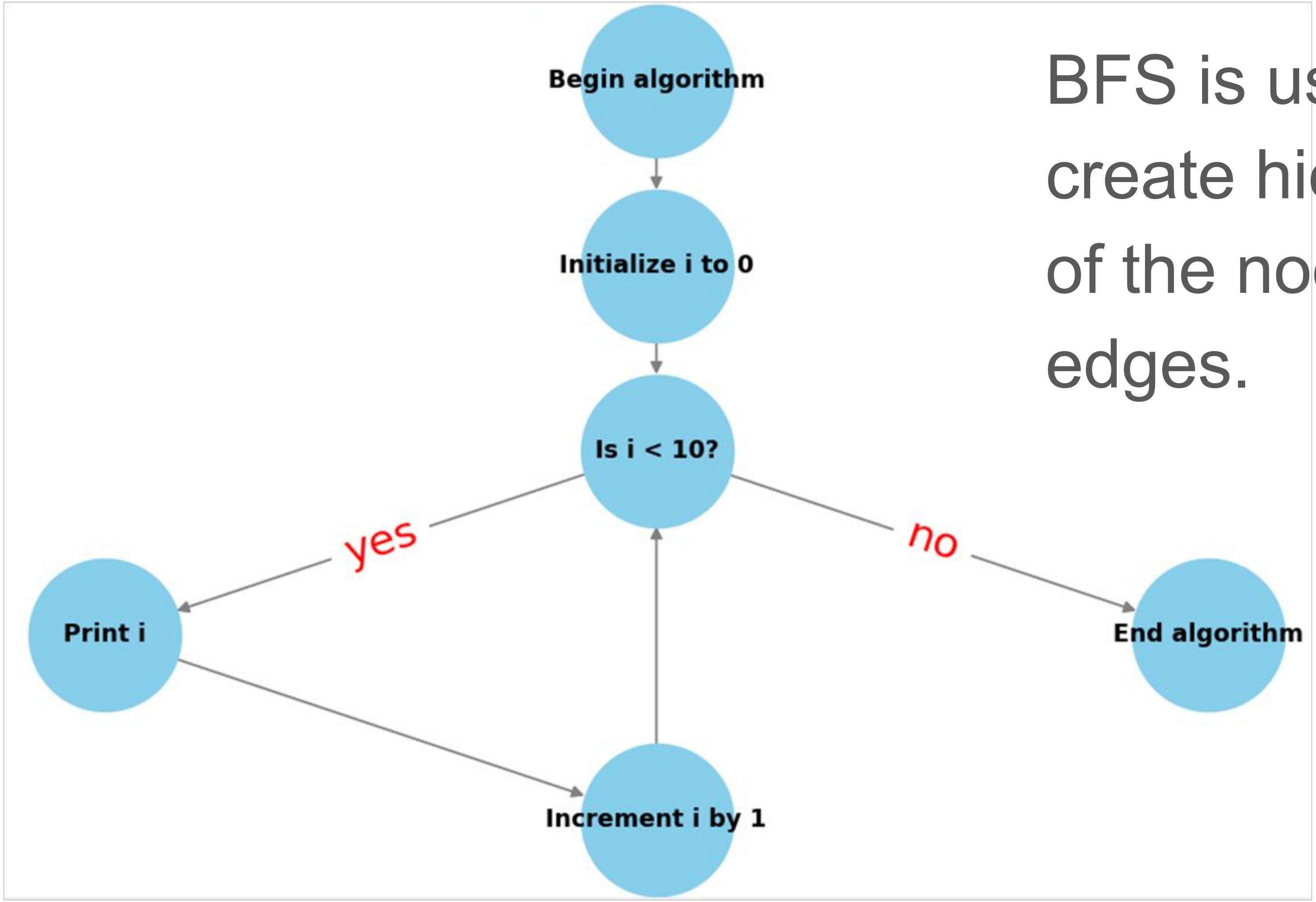
A simple loop to display the 10 digits which is handwritten.

Handwritten data to json data format using OCR (Optical Character Recognition) and Computer vision


```json
{
    "nodes": [
      {"id": "start", "type": "start", "text": "Begin algorithm"},
      {"id": "n1", "type": "process", "text": "Initialize i to 0"},
      {"id": "n2", "type": "decision", "text": "Is i < 10?"},
      {"id": "n3", "type": "process", "text": "Print i"},
      {"id": "n4", "type": "process", "text": "Increment i by 1"},
      {"id": "end", "type": "end", "text": "End algorithm"}
    ],
    "edges": [
      {"from": "start", "to": "n1"},
      {"from": "n1", "to": "n2"},
      {"from": "n2", "to": "n3", "condition": "yes"},
      {"from": "n3", "to": "n4"},
      {"from": "n4", "to": "n2"},
      {"from": "n2", "to": "end", "condition": "no"}
    ]
}
```



The json data file is converted into nodes and edges. It is currently unorganized.



BFS is used to create hierarchy of the nodes and edges.

Diagram score is only 40/100 when using only LLM

```
(myenv) C:\Users\aster\sem7\Capstone>python analyze_diagram.py
Semantic Score: 40
No, the student's flowchart is incomplete and does not correctly represent the logic for printing numbers from 0 to 9 in a loop.
 The crucial missing step is the action to be performed *within* the loop before the increment.  The sequence should be: Initiali
ze loop -> Condition check -> **Print the number** -> Increment -> Condition check (and loop back). As it stands, the print state
ment happens *after* the loop has potentially already incremented past 9, leading to incorrect output or an infinite loop dependi
ng on the condition check.

Score: 40/100

Feedback: The logic could be improved. Ensure that your steps match the intended algorithm more closely.
```

Diagram score is 72/100 when graph analysis verifies the structure.

```
(myenv) C:\Users\aster\sem7\Capstone>python analyze_diagram_with_graph_llm.py
Structural Score: 100
Semantic Score: 30
Final Combined Score: 72.0
No, the student's flowchart does not correctly represent the logic of printing numbers from 0 to 9 in a loop.  The problem is tha
t the "Is i < 10?" condition check is only performed *once*, before the print statement.  The algorithm will print i (which is 0)
 only once and then terminate.  A proper loop requires the condition check to be placed *before* the print and increment steps, a
nd the flow should return to the condition check after incrementing.  The code will execute the print and increment once.  To cre
ate a loop it needs to loop back to the conditional.
```

Methodology:

Step	Method	Impact
Graph-Based Diagram Representation and Traversal	Convert diagrams into **graph structures** where nodes represent components and edges represent flow. Use **graph traversal algorithms** to check the **logical flow** and consistency.	Ensures full validation of the **logical flow** and identifies missing connections or incorrect branches, improving accuracy by **30-40%**.
Error Detection and Structural Validation	Implement **error detection** to identify **structural mistakes** like missing branches, unconnected components, or faulty loops. Ensure logical completeness of all nodes and flows.	Improves detection of critical structural errors, enhancing grading accuracy for complex diagrams by **20-30%**.
LLM-Based Contextual Understanding	Use **LLMs** to analyze the **semantic content** of text within diagram blocks, ensuring the text is contextually correct and aligned with the problem requirements.	Enhances the evaluation of **textual content** and **contextual relevance**, adding a **10-15% boost** in semantic accuracy.

USE CASES

1. **Business Process Automation**: Automating workflows and decision-making processes using flowcharts, helping businesses improve efficiency.
2. **Educational Tools**: Assisting students in understanding complex concepts through interactive flowchart-based learning materials.
3. **Software Development**: Assisting developers in debugging, testing, and designing complex systems using flowcharts.
4. **Healthcare Decision Support**: Aiding doctors with medical flowcharts for diagnosing and recommending treatments.
5. **Process Modeling**: Streamlining industrial or engineering process modeling for optimization.

References:

1. **Rajlaxmi Patil, Aditya Ashutosh Kulkarni, Ruturaj Ghatage, Sharvi Endait, Dr. Geetanjali Kale, Raviraj Joshi.** "Automated Assessment of Multimodal Answer Sheets in STEM." arXiv preprint arXiv:2409.15749, 2024. Available: https://arxiv.org/abs/2409.15749
2. **C. David Betancourt Montellano, C. Onder Francisco Campos Garcia, Roberto Oswaldo Cruz Leija.** "Recognition of Handwritten Flowcharts using Convolutional Neural Networks." *International Journal of Computer Applications*, Vol. 184, Issue 1, 2022. Available: https://www.ijcaonline.org/archives/volume184/number1/2921969-2022921969
3. **Martin Bresler, Daniel Pruša, Vaclav Hlavac.** "Online Recognition of Sketched Arrow-Connected Diagrams." *International Journal on Document Analysis and Recognition (IJDAR)*, Vol. 19, 2016. Available: https://link.springer.com/article/10.1007/s10032-016-0269-z
4. **https://arxiv.org/html/2406.10057v1**

Simple Business overview:

Aspect	Description
Value Proposition	Provides **accurate, automated grading** of diagrams using **graph-based logic** and **LLM evaluation**, supporting **multiple valid answers** and giving **instant feedback** to enhance student learning.
Target Market	**Educational institutions**, **e-learning platforms**, and **certification bodies** needing scalable, accurate STEM assessment tools.
Revenue Model	**SaaS subscriptions**, **pay-per-assessment fees**, and **licensing** to EdTech platforms with enterprise solutions for large institutions.

Phase	Duration	Milestone/Task	Goals
Phase 1: Research & Planning	**Month 1 - 2**	Conduct **literature review**, finalize **project scope**, and design **system architecture**.	Define **objectives, scope**, and project blueprint.
Phase 2: Graph-Based Module	**Month 3 - 5**	Develop **graph-based analysis module** for **logic validation** and test with simple diagrams.	Ensure **graph-based representation** and **logical flow detection**.
Phase 3: LLM Module	**Month 6 - 8**	Develop **LLM module** for evaluating **semantic content** in diagrams and test on sample data.	Achieve high accuracy in **textual and contextual evaluation**.
Phase 4: Integration & Errors	**Month 9 - 10**	Integrate **graph and LLM modules**, implement **error detection**, and test on real-world diagrams.	Create a **unified system** with improved **accuracy and error detection**.
Phase 5: Testing & Optimization	**Month 11 - 12**	Conduct **extensive testing** on various diagram types, optimize, and finalize the system for deployment.	Ensure **high accuracy, performance**, and readiness for **deployment**.