Pusat Sains Matematik UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH	**SUBJECT: BSD2513** **ARTIFICIAL INTELLIGENCE** **TOPIC:** Chapter 1: Introduction to Artificial Intelligence Chapter 2: Search Algorithms **LAB REPORT 1**	**MARKS**: 25(5%)

CLO	Description	PLO Mapping	Percentage	Marks
CLO1	Acquire the artificial intelligence concepts and methodologies in data science.	PLO1: Knowledge and Understanding C3: Application	1%	5
CLO2	Demonstrate critical thinking ideas of artificial intelligence knowledge in problem-solving situation.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills CLO3: Application	1%	5
CLO3	Develop an artificial intelligence system prototype using appropriate software.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	3%	15

Name: Nur Safiyyah binti Sulaiman
ID: SD23049

Laboratory Report Objectives

By the end of this lab, students should be able to:

1. articulate AI capability types and their relevance to real deployments;
2. reason about and trace Breadth-First Search (BFS) and Depth-First Search (DFS) on given graphs; and
3. implement a minimal, reliable web app (Streamlit or similar) that runs these traversals, explains their complexity, and is reproducible from a GitHub repository (with a live deployment link).

CASE STUDY:

Modern organizations (e.g., logistics, telecom, and public services) routinely face search and routing problems under time and resource constraints. Artificial Intelligence (AI) methods, particularly search algorithms where are used to explore large state spaces to find feasible or optimal solutions. Typical use cases include:

- Network traversal & connectivity: discovering reachable components or shortest hops across campus or city networks.
- Work allocation & incident response: exploring task graphs to sequence actions under precedence constraints.
- Navigation and routing: traversing road graphs to plan visits while honouring constraints (e.g., fuel, time windows).

Question 1 General Knowledge

Explain types of AI by capability and functionality. For each capability and functionality, give one realistic application example in data-driven operations. Conclude with a short note on ethical and governance considerations relevant to deployment (e.g., transparency, safety, bias).

There are three types of AI based on capabilities which is Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI), and Artificial Super Intelligence (ASI). At ANI stage, the machine does not possess any thinking ability and it just performs a set of pre-defined functions. For example, supply chain optimization system that uses data from sales and inventory to predict product demand and plan deliveries. Next, in AGI, machines will possess the ability to think and make decisions just like us humans such as Global Financial AI Advisor the and last one, ASI is the stage of AI when the capability of computers will surpass human beings. For example, AI-driven enterprise management system that controls the entire global supply chain in real time. Besides, for functionality, there are 4 types which is reactive machines AI, limited memory AI, theory of mind AI, and self-aware AI. The example for reactive machine AI is IBM Chess program, for limited memory AI is self-driving cars use sensors to identify civilians crossing the road and traffic signals to make better driving decisions and for theory of mind AI is advanced customer service assistant that can detect a user's mood through voice tone and lastly is for self-aware AI is self-managing AI system.

In conclusion, ethical and governance considerations including safety, fairness, and transparency are crucial to preventing unfair or discriminatory outcomes and ensuring that AI is applied sensibly and for everyone's benefit.

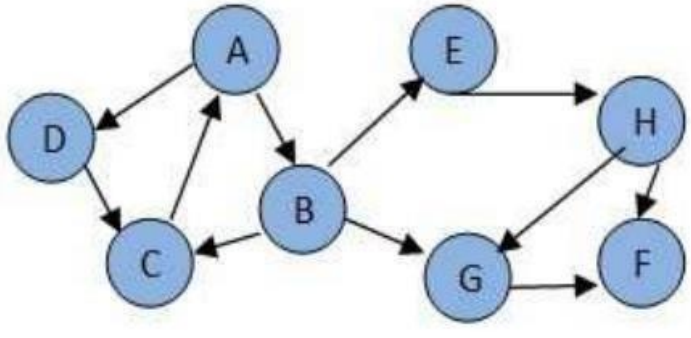
(5 Marks)
CO1 PO1)

Question 2 Python: Search Algorithms

Consider the directed graphs provided below. When expanding nodes, assume alphabetical tie-breaking, i.e., when multiple adjacent nodes are available, always visit the node with the name closest to the beginning of the alphabet first using:

a. breadth-first search (BFS) and,
b. depth-first search (DFS).

Discuss the level and the process path clearly.



```
from collections import deque
# Graph adjacency list based on your image
graph = {
    'A': ['B', 'D'],
    'B': ['C', 'E', 'G'],
    'C': ['D'],
    'D': ['A'],
    'E': ['H'],
    'F': [],
    'G': ['F', 'H'],
    'H': []
}

# Breadth-First Search (BFS) with alphabetical ordering
def bfs(start):
    visited = set()
    queue = deque([start])
    order = []

    while queue:
        node = queue.popleft()
        if node not in visited:
            visited.add(node)
            order.append(node)
            # Add neighbors in alphabetical order to queue
            for neighbor in sorted(graph[node]):
                if neighbor not in visited:
                    queue.append(neighbor)
    return order

# Depth-First Search (DFS) with alphabetical ordering
def dfs(start):
    visited = set()
    stack = [start]
    order = []

    while stack:
        node = stack.pop()
        if node not in visited:
            visited.add(node)
            order.append(node)
            # Add neighbors in reverse alphabetical order so smallest is popped first
            for neighbor in sorted(graph[node], reverse=True):
                if neighbor not in visited:
                    stack.append(neighbor)
    return order

# Run the searches starting from 'A'
```

```
bfs_order = bfs('A')
dfs_order = dfs('A')

print("BFS Order:", bfs_order)
print("DFS Order:", dfs_order)
```

```
BFS Order: ['A', 'B', 'D', 'C', 'E', 'G', 'H', 'F']
DFS Order: ['A', 'B', 'C', 'D', 'E', 'H', 'G', 'F']
```

(5 Marks)
(CO2PO2)

Breadth-first search (BSF)

Go to B and D, which is A's neighbors. Then, examine B first because it comes before D. After that go to B's neighbors, C, E, and G. They are added to the list to be visited later in alphabetical order. Next, we go to D, but as its neighbor A has already been visited, we don't go there. After seeing neighbor D, we proceed to visit C. E comes next, which takes us to H. Next is G, which links to both F and H (although H is already on the list). Lastly, we go to H, followed by F.

Depth-first search

Proced to B first, starting at A. Then, descend as far as we can from B to C. After that, move to D from C. Attempt to reach A from D, but because it has already been reached, turn around. Return to B and then go to E and proceed from E to H. Next, go back to B after completing H, and then we go to G. Lastly, go from G to F.

(15 Marks)
(CO3PO3)

Save your work in both .py and PDF formats. Name your files using the following format: StudentID_LabX. Submit both files through the Kalam platform by 10th October 2025, 11:59 PM. In addition, deploy your Streamlit application and include the public URL to your deployed app and GitHub repository link inside your report. Late submissions will only be considered with prior approval.

Pusat Sains Matematik	**SUBJECT: BSD3513** **INTRODUCTOION TO ARTIFICIAL INTELLIGENCE**	**MARKS**: 25(5%)
	TOPIC: Chapter 1: Foundations of Artificial Intelligence Chapter 2: Search Algorithms	
	LAB REPORT 1	

NAME: **STUDENT ID:** **SECTION:**

Mark for CO1: /5

Rubric for CO2.

Instruction: For CO2, assess each item using the given scales.

Demonstrate critical thinking ideas of artificial intelligence knowledge in problem-solving situation.								
Item Assessed (Cognitive)	**Very Poor 0**	**Poor 1**	**Fair 2**	**Good 3**	**Very Good 4**	**Excellent 5**	**Weightage**	**Score**
Apply and analyse relevant artificial intelligence knowledge.	The work has not done.	Poorly applied and analysed relevant artificial intelligence knowledge and results.	Applied and analysed relevant artificial intelligence knowledge but failed to achieve successful results.	Applied and analysed relevant artificial intelligence knowledge but arrive at satisfactory results.	Applied and analysed relevant artificial intelligence knowledge to arrive at successful results.	Applied and analysed relevant artificial intelligence knowledge to arrive at excellent results.	0.5	

Using logical, rational or problem-solving appropriate to the artificial intelligence problems.	The work has not done.	The work needs to demonstrate logical, rational or problemsolving understanding appropriate to the artificial intelligence problems.	The work has demonstrated some logical, rational or problem-solving understanding appropriate to the artificial intelligence problems.	The work has demonstrated logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	The work has demonstrated a thorough logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	The work has demonstrated a thorough and classy logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	0.5	
						Total Score	**1**	**/5**

Rubric for CO3.

Instruction: For CO3, assess each item using the given scales.

CO3: Develop an artificial intelligence system prototype using appropriate software.							
Item Assessed (Cognitive)	**Very Poor 0**	**Poor 1**	**Fair 2**	**Good 3**	**Very Good 4**	**Excellent 5**	**Score**
Utilizing the appropriate tools / software effectively	No relevant tool used.	Tools used but did not enhance solution or information clarity.	Tools used but with limited enhancement; minimal functionality demonstrated.	Tools used appropriately to produce a functional solution with clear output.	Tools used effectively to enhance clarity, performance, and solution quality.	Tools used optimally with advanced features, clear design, and effective interaction to display the solution.	
Code functionality, clarity & structure	No code constructed.	Code incomplete or mostly nonfunctional; unclear and poorly structured.	Partially functional code; errors present; structure somewhat difficult to follow.	Mostly functional code with minor errors; clear structure and readable.	Fully functional and well-structured code; clearly commented and readable.	Fully functional, optimized, modular, and welldocumented code; demonstrates best practices.	
Deployment & Version Control (GitHub + Streamlit or etc)	No deployment and no GitHub repository.	GitHub repo exists but incomplete OR app deploy attempt failed.	GitHub repo available with basic files; deployment page exists but app not functioning correctly.	Working deployment provided; GitHub repo contains main code files.	Working deployment with complete repository (README, code, requirements); clearly accessible.	Fully deployed app with professional GitHub repo (README, screenshots, instructions, modules, tags); live Streamlit app runs smoothly and reliably.	

	Total Score	**/15**