|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA  **KHOA Công Nghệ Thông Tin**  BỘ MÔN: Công Nghệ Phần Mềm |

**ĐỀ THI VÀ BÀI LÀM**

Tên học phần: Trí tuệ nhân tạo

Mã học phần: Hình thức thi: *Tự luận có giám sát*

Đề số: **01** Thời gian làm bài: 75 phút *(không kể thời gian chép/phát đề)*

Được sử dụng tài liệu khi làm bài.

**Họ tên:** ……………………………**Lớp**:……………………………**MSSV**:………………………

Sinh viên làm bài trực tiếp trên tệp này, lưu tệp với định dạng MSSV\_HọTên.pdf và nộp bài thông qua MSTeam:

***Câu 1*** (5 *điểm*): Cho đồ thị vô hướng G = (V,E) như hình vẽ với V là tập đỉnh và E là tập cạnh.

Diagram

Description automatically generated

1. *(1 điểm)* Hãy viết đoạn code biểu diễn đồ thị trên bằng cách khởi tạo tập đỉnh V và tập cạnh E.   
   (Ví dụ: V = [“S”, “A”, “B”], E =[(“S”, “A”), (“S”, “B”)])

|  |
| --- |
| **# Trả lời:** Dán code vào bên dưới  V = ["S", "A", "B", "C", "D", "E", "F", "G", "H"]  E = [("S", "A"), ("S", "B"), ("S", "C"), ("A", "B"), ("A", "D"), ("B", "C"), ("B", "D"), ("B", "F"), ("B", "G"), ("C", "F"), ("D", "E"), ("E", "F"), ("E", "G"), ("F", "H"), ("H", "G")]  graph = {}  for vertex in V:  graph[vertex] = []  for edge in E:  u= edge[0]  v = edge[1]  graph[u].append(v)  graph[v].append(u)  print("Danh sách đỉnh và các đỉnh kề tương ứng:")  for vertex in V:  print(f"{vertex}: {graph[vertex]}") |

1. *(3 điểm)* Hãy viết chương trình sử dụng thuật toán **tìm kiếm theo chiều rộng (BFS)** để tìm đường đi từ đỉnh “S” đến đỉnh “G” trong đồ thị được biểu diễn ở câu a). Trong chương trình, hãy in ra thứ tự đỉnh khám phá trong quá trình tìm kiếm. Nếu không tìm thấy thì in “*Khong tim thay duong di*”

|  |
| --- |
| **# Trả lời:** Dán code vào bên dưới  V = ["S", "A", "B", "C", "D", "E", "F", "G", "H"]  E = [("S", "A"), ("S", "B"), ("S", "C"), ("A", "B"), ("A", "D"), ("B", "C"), ("B", "D"), ("B", "F"), ("B", "G"), ("C", "F"), ("D", "E"), ("E", "F"), ("E", "G"), ("F", "H"), ("H", "G")]  graph = {}  for vertex in V:  graph[vertex] = []  for edge in E:  u= edge[0]  v = edge[1]  graph[u].append(v)  graph[v].append(u)  print("Danh sach cac dinh ke tuong ung:")  for vertex in V:  print(f"{vertex}: {graph[vertex]}")  def BFS(graph,start,goal):  visited = set()  queue = [start]  visited.add(start)  explored = []  while queue:  status = queue.pop(0)  explored.append(status)  if status == goal:  return explored  for neighbor in graph[status]:  if neighbor not in visited:  visited.add(neighbor) # đánh dấu  queue.append(neighbor)    return False  result = BFS(graph,"S","G")  if result:  print("thu tu dinh kham pha:")  print(result)  else:  print("khong tim thay duong di")  **# Trả lời:** Dán kết quả thực thi vào bên dưới: |

1. *(1 điểm)* Hãy trực quan hóa kết quả tìm kiếm đường đi từ đỉnh “S” đến đỉnh “G”

|  |
| --- |
| **# Trả lời:** Dán code vào bên dưới    **# Trả lời:** Dán kết quả thực thi vào bên dưới: |

***Câu 2*** (3 *điểm*):

1. *(2 điểm)* Mô tả thuật toán hoặc hàm thực thi chiến lược Minimax

|  |
| --- |
| **# Trả lời:** viết mô tả thuật toán hoặc dán code vào bên dưới |

1. *(1 điểm)* trình bày ứng dụng của thuật toán Minimax

|  |
| --- |
| **# Trả lời**: viết câu trả lời vào bên dưới |

***Câu 3***(2 *điểm*): Cho hàm f(*x*) =

1. *(2 điểm)* Viết chương trình tính giá trị lớn nhất của hàm f(x) sử dụng thuật toán Gradient Descent

|  |
| --- |
| **# Trả lời**: viết câu trả lời vào bên dưới  **# Trả lời**: Dán kết quả thực thi vào bên dưới |

Đà Nẵng, ngày 27 tháng 3 năm 2024

|  |  |
| --- | --- |
| **GIẢNG VIÊN BIÊN SOẠN ĐỀ THI** | **TRƯỞNG BỘ MÔN** |
|  | (đã duyệt) |