

Đề 4.


Câu 1:

Photos - 262065724_629420771392029_3224335924468646291_n.png 162%

See all photos + Add to

🔍 🔍 🗑️ ❤️ ↺️ ✂️ Edit & Create ▾ 📁 Share 🖨️ ...

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.



a) (2 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

b) (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 "Không tìm thấy đường đi"

```
def BFS(initialState, goal):
    frontier = [initialState]
    explored = []

    while frontier:
        state = frontier.pop(0)
        explored.append(state)
        if state == goal:
            return explored
        for neighbor in graph[state]:
            if neighbor not in frontier + explored:
                frontier.append(neighbor)
    return False

if __name__ == "__main__":
    V = ['S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
    E = [('S', 'A'), ('S', 'B'), ('S', 'C'),
          ('A', 'D'), ('A', 'B'),
          ('B', 'D'), ('B', 'F'), ('B', 'G'),
          ('C', 'F'),
          ('D', 'E'),
          ('E', 'G'), ('E', 'F'),
          ('F', 'H'),
          ('G', 'H')]
    graph = {}
    for node in V:
        graph[node] = []
```

```

for edge in E:
    firstNode = edge[0]
    secondNode = edge[1]

    firstList = graph[firstNode]
    firstList.append(secondNode)
    graph[firstNode] = firstList

    secondList = graph[secondNode]
    secondList.append(firstNode)
    graph[secondNode] = secondList
result = BFS('S', 'G');
if result:
    print("explored: ", result)
else:
    print('Không tìm thấy đường đi');

```

Câu 2.

Photos - 262322528_1734925803370401_8984446350371153385_n.png 101%

See all photos + Add to

sẽ

a. (1 điểm) Mô tả thuật toán hoặc hàm thực thi thuật toán k -means
Trả lời: (viết mô tả thuật toán hoặc code vào bên dưới)

b. (1 điểm) Nếu sử dụng thuật toán k -means với $k = 2$ thì kết quả phân nhóm sẽ như thế nào? (các điểm thuộc mỗi nhóm, trọng tâm của mỗi nhóm)
Trả lời: (viết câu trả lời vào bên dưới)

Type here to search

70°F 2:19 PM 11/29/2021

a. Mô tả thuật toán:

B1: Khởi tạo ngẫu nhiên các trọng tâm

B2: Nhóm dữ liệu: tính khoảng cách giữa các điểm và các tâm. Sau đó xếp chúng vào một cụm sao cho khoảng cách từ điểm đó tới tâm của cụm là nhỏ nhất.

B3: Cập nhật trọng tâm

Nếu không có sự thay đổi thì dừng thuật toán

B4: Quay lại B2

- b. Cụm 1: trọng tâm (2.5, 0) gồm hai điểm (2,0) (3,0)
Cụm 2: trọng tâm (0.5, 3.5) gồm hai điểm (0,3) (1,4)

Câu 3:

```
import math

def grad(x):
    return 2 * (math.exp(2*x) - 4/math.exp(2*x))

def cost(x):
    temp = math.exp(x) - 2/math.exp(x)
    return pow(temp, 2)

def myGD(eta, x0):
    x = [x0]
    for it in range(100):
        x_new = x[-1] - eta*grad(x[-1])
        if abs(grad(x_new)) < 1e-3:
            break
        x.append(x_new)
    return (x, it)

if __name__ == "__main__":
    (x1, it1) = myGD(.1, 0.2)
    print('Solution x1 = %f, cost = %f, obtained after %d iterations'%(x1[-1], cost(x1[-1]), it1))
```