

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

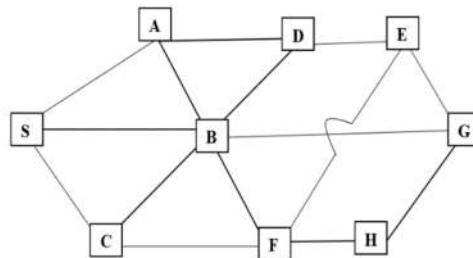
Mã học phần: Hình thức thi: *Tự luận có giám sát*

Thời gian làm bài: 70 phút (không kể thời gian chép/phát đề)

Được sử dụng tài liệu khi làm bài.

MSSV: 102180083

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.



- (Ví dụ: $V = [“S”, “A”, “B”]$, $E = [(“S”, “A”), (“S”, “B”)]$)

```
V = ["S", "A", "B", "C", "D", "F", "E", "H", "G"]
E = [
    ("S", "A"), ("S", "B"), ("S", "C"),
    ("A", "D"), ("A", "B"),
    ("C", "B"), ("C", "F"),
    ("B", "D"), ("B", "F"),
    ("D", "E"),
    ("F", "E"), ("F", "H"),
    ("E", "G"),
    ("H", "G")
]
```

- 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*”

Trả lời: Chương trình

```
def BFS(initialPoint, endPoint):
    frontier = [initialPoint]
    explored = []
```

```

while frontier:
    # Using array instead of queue
    point = frontier.pop(0)
    explored.append(point)

    if point == endPoint:
        return explored

    for edge in E:
        if edge[0] == point and (edge[1] not in explored and edge[1] not in
frontier):
            frontier.append(edge[1])

    return False

result = BFS("S", "G")
if result:
    print(result)
else:
    print("Khong tim thay duong di")

```

Trả lời: Kết quả thực thi:

```
['S', 'A', 'B', 'C', 'D', 'F', 'E', 'H', 'G']
```

Câu 2 (2 điểm): Cho không gian Oxyz với 6 điểm có tọa độ tương ứng (0,4,1), (1,3,3) (4,0,2),(3,1,4),(2,1,2) và (2,3,4).

a) (1 điểm) Mô tả thuật toán hoặc hàm thực thi thuật toán k -means

Trả lời: viết mô tả thuật toán hoặc dán code vào bên dưới

```

import copy
import numpy as np

def initiateCenter(X: np.ndarray, K: np.ndarray) -> np.ndarray:
    return X[np.random.choice(X.shape[0], K, replace=False)]

def update_K_points(X: np.ndarray, K: np.ndarray, labels: np.ndarray):
    for K_id in range(K.shape[0]):
        cluster = X[K_id == labels]

        clusterX = cluster[:, 0]
        clusterY = cluster[:, 1]
        clusterZ = cluster[:, 2]

        K[K_id] = np.array([np.mean(clusterX), np.mean(clusterY), np.mean(clusterZ)])

def clustering(X: np.ndarray, K: np.ndarray) -> np.ndarray:
    labels = np.zeros(X.shape[0])

```

```

for X_id in range(X.shape[0]):
    K_closest_id = np.argmin(np.sum((K - X[X_id])**2, 1)) # Find the K point nearest by L2 Norm
    labels[X_id] = K_closest_id

return labels

def run_K_means(X: np.ndarray, K: np.ndarray) -> tuple:

    while True:
        # Save a copy of previous K_points
        K_before = copy.deepcopy(K)

        # Labeling for the input
        labels = clustering(X, K)

        # After labeling, update K_point in the center
        update_K_points(X, K, labels)

        # Check if the previous K_points equal to the new K_point
        if (np.all(K_before == K)):
            return (K, labels)

# Initiating
X = np.array([[0,4,1], [1,3,3], [4,0,2], [3,1,4], [2,1,2], [2,3,4]], np.float32)

K = 2
K_points = initiateCenter(X, K)

print(X)
print(K_points)

print("-----K-Means-----")
(labels, K_points) = run_K_means(X, K_points)
print(X)
print(K_points)
print(labels)

```

- 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).

Nhóm 1: Trọng tâm: [0.5, 3.5, 2.0]

Các điểm thuộc nhóm 1:

[4.0, 0.0, 2.0]

[3.0, 1.0, 4.0]

[2.0, 1.0, 2.0]

[2.0, 3.0, 4.0]

Nhóm 2: Trọng tâm: [2.75, 1.25, 3.0]

Các điểm thuộc nhóm 1:

[0.0, 4.0, 1.0]

[1.0, 3.0, 3.0]

Câu 3(3 điểm): Cho hàm $f(x) = \left(e^{-x} - \frac{4}{e^{2x}}\right)^2$

a) (1 điểm) Khai triển đạo hàm cấp 1 của $f(x)$

Trả lời: Khai triển và kết quả đạo hàm ở bên dưới

$$2 * \left(e^{-x} - \frac{4}{e^{2x}}\right) * \left(-e^{-x} + \frac{8}{e^{2x}}\right) = 2 * f(x) * \left(-e^{-x} + \frac{8}{e^{2x}}\right)$$

b) (2 điểm) Viết chương trình tính giá trị nhỏ nhất của $f(x)$ sử dụng thuật toán Gradient Descent

Trả lời: viết câu trả lời vào bên dưới

```
import numpy as np

def f(x):
    return (np.exp(-x) - 4 / np.exp(2*x)) ** 2

def f1(x):
    return 2 * f(x) * (-np.exp(-x) + 8 / np.exp(2*x))

x = 2
learning_rate = 0.01
tolerance = 1e-10

x_new = x
while True:
    x_new = x_new - learning_rate * f1(x)

    if (np.abs(f1(x_new)) < tolerance):
        break

print(x)
print(x_new)
```

Trả lời: Dán kết quả thực thi vào bên dưới

2
1.3863506264647967

GIẢNG VIÊN BIÊN SOẠN ĐỀ THI

Đà Nẵng, ngày 26 tháng 11 năm 2021
TRƯỞNG BỘ MÔN
(đã duyệt)

