

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: **0004** Thời gian làm bài: 70 phút (*không kể thời gian chép/phát đề*)

Được sử dụng tài liệu khi làm bài.

Họ tên: Văn Hữu An **Lớp:** 19TCLC_DT3 **MSSV:** 102190101

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 (2 điểm): Cho không gian Oxy với các điểm có tọa độ tương ứng $(1.5, 6)$, $(3, 4.5)$, $(2.25, 6)$, $(6, 1.5)$, $(7.5, 3)$, $(6, 7.5)$, $(4.5, 9)$ và $(4.5, 6)$.

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

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K .

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y .

Bước 1. Chọn K điểm bất kỳ làm các center ban đầu.

Bước 2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.

Bước 3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

Bước 4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

Bước 5. Quay lại bước 2.

b) (1 điểm) Nếu sử dụng thuật toán k -means với $k = 3$ 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

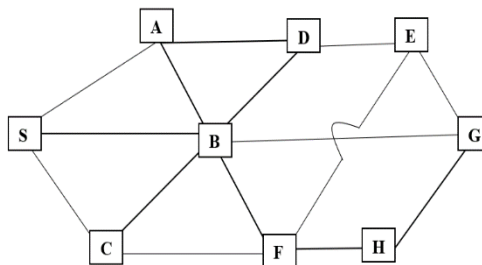
Với $k = 3$, chúng ta có 3 nhóm với 3 trọng tâm như sau:

Trọng tâm 1 $(5.00, 7.50)$ có các điểm thuộc nhóm là $(4.5, 9)$, $(6, 7.5)$ và $(4.5, 6)$.

Trọng tâm 2 $(6.75, 2.25)$ có các điểm thuộc nhóm là $(6, 1.5)$ và $(7.5, 3)$.

Trọng tâm 3 $(2.25, 5.5)$ có các điểm thuộc nhóm là $(1.5, 6)$, $(2.25, 6)$ và $(3, 4.5)$.

Câu 2 (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

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

```
E = [[0,1,1,1,0,0,0,0,0],
```

```
      [1,0,1,0,1,0,0,0,0],
```

```
      [1,1,0,1,1,0,1,0,1],
```

```
      [1,0,1,0,0,0,1,0,0],
```

```
      [0,1,1,0,1,0,0,0,0],
```

```
      [0,0,0,0,1,0,1,0,1],
```

```
      [0,0,1,1,0,1,0,1,0],
```

```
      [0,0,0,0,0,0,1,0,1],
```

```
      [0,0,1,0,0,1,0,1,0]]
```

E là ma trận MxM với M là số đỉnh, có thứ tự tương ứng với thứ tự các đỉnh trong V. Với mỗi $E[x,y]$

là đường đi từ đỉnh thứ x đến đỉnh thứ y, bằng 1 nếu có đường đi và bằng 0 nếu không có đường đi

hoặc đi đến chính nó.

ví dụ $E[3,6] = 1$ là đường đi từ đỉnh thứ 3(‘C’) đến đỉnh thứ 6(‘F’) có thể đi được,

$E[0,0]$ là đường đi từ đỉnh thứ 0(‘S’) đến chính nó là không thể đi đ

- b) (3 điểm) Hãy viết chương trình sử dụng thuật toán **tìm kiếm theo chiều sâu (DFS)** để tìm đường đi từ đỉnh “S” đến đỉnh “H” 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: Dán code vào bên dưới

```
start = 0
```

```
open = [start]
```

```
close = []
```

```
success = False;
```

```
count = 0
```

```
dinhcha = {}
```

```
while len(open)>=1 :
```

```
    count += 1
```

```
    O = open.pop(0)
```

```
    close.append(O)
```

```
    if O == 8:
```

```
        success = True
```

```
        break
```

```
    khampha = []
```

```
    for y in range(0,9):
```

```
        if E[O][y] == 0:
```

```
            continue
```

```
        if y in close or y in open:
```

```
            continue
```

```
        khampha.append(y)
```

```
    dinh = []
```

```
    for i in khampha:
```

```
        dinh.append(V[i])
```

```
        dinhcon = i
```

```
        open.insert(0, dinhcon)
```

```
        dinhcha[dinhcon] = O
```

```
    print("Kham pha ra dinh: ",dinh)
```

```
if success == False:
```

```
    print("Không tìm thấy đường đi")
```

```
else:
```

```

truyvet = [V[O]]
while dinhcha.get(O)!=None:
    O = dinhcha.get(O)
    truyvet.append(V[O])
truyvet.reverse()
print("Duong di: ")
print(*truyvet)
# Trả lời: Dán kết quả thực thi vào bên dưới:
Kham pha ra dinh: ['A', 'B', 'C']
Kham pha ra dinh: ['F']
Kham pha ra dinh: ['E', 'H']
Kham pha ra dinh: ['G']
Duong di:
S C F H G

```

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

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

```

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

$$f(x) = \left(e^{2x} - \frac{4}{e^{-x}}\right)^2 = (e^{2x} - 4e^x)^2 = (e^{4x} - 8e^{3x} + 16e^{2x})$$


$$f'(x) = (e^{4x} - 8e^{3x} + 16e^{2x})' = (4e^{4x} - 24e^{3x} + 32e^{2x})$$


```

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 math
def f(x):
    return math.exp(4*x) - 8*math.exp(3*x) + 16*math.exp(2*x)
def gradient(x):
    return 4*math.exp(4*x) - 24*math.exp(3*x) + 32*math.exp(2*x)
def GD():
    x = 0
    gd = 0.2
    learningRate = 0.1
    while abs(gd) > 0.0001:
        gd = gradient(x)
        x = x - learningRate*gd
    print("x= ",x)
    print("min f(x) = ",f(x))
GD()
# Trả lời: Dán kết quả thực thi vào bên dưới
x= -6.337391478159672
min f(x) = 5.0020437555843485e-05

```

(đã duyệt)