

BỘ XÂY DỰNG
TRƯỜNG ĐHXD MIỀN TÂY

ĐÁP ÁN – THANG ĐIỂM
KỲ THI KTHP HỌC KỲ I NĂM HỌC 2024-2025

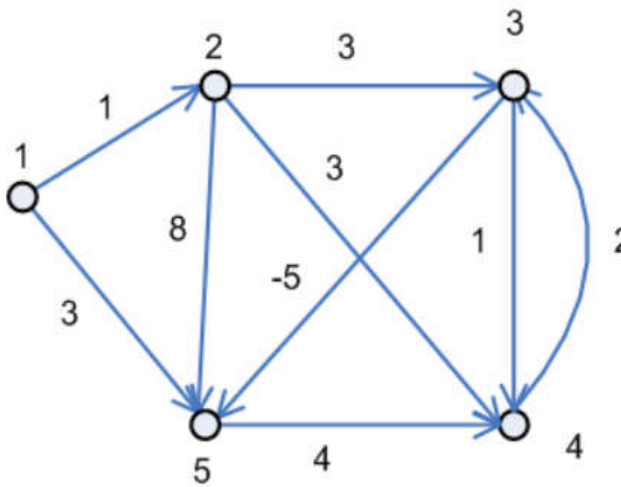
Trình độ: ĐẠI HỌC; Ngày thi: 03/01/2025

Môn: LÝ THUYẾT ĐỒ THỊ

ĐÁP ÁN ĐỀ THI CHÍNH THỨC

(Đáp án - thang điểm gồm 04 trang)

Câu	Đáp án	Nội dung	Thang điểm
I		TRẮC NGHIỆM	5.0
1	D	Không có khuyên và không có cạnh lặp	0.25
2	B	Một số chẵn	0.25
3	C	Bậc bằng 1	0.25
4	A	G1 và G2 không đẳng cấu với nhau	0.25
5	D	Tất cả các đỉnh đều bậc chẵn	0.25
6	C	33	0.25
7	A	Ngăn xếp (Stack)	0.25
8	D	$n*(n-1)/2$	0.25
9	C	$m*n$	0.25
10	B	n	0.25
11	C	Luôn tìm được đường đi giữa hai đỉnh bất kỳ	0.25
12	D	Queue	0.25
13	A	Thuật toán Floyd	0.25
14	A	Đồ thị đầy đủ	0.25
15	D	3 màu	0.25
16	D	Phân loại thành phần liên thông	0.25
17	C	Chẵn	0.25
18	C	BFS hoặc DFS	0.25
19	B	Có đường đi Euler	0.25
20	D	Thuật toán Brute-Force	0.25

II	TỰ LUẬN	5.0
Câu 1	Trình bày Nguyên lý duyệt Đồ thị vô hướng theo chiều sâu (Depth First Search – DFS), Cài đặt thuật toán DFS bằng phương pháp Đệ quy, Vẽ lưu đồ thuật toán (1.5 điểm)	1.5
Câu 2	Dùng ngôn ngữ lập trình Python để hiện thực Thuật toán DFS ở Câu 1 : bằng phương pháp Đệ quy (1.0 điểm)	1.0
Câu 3	Cho đồ thị như hình sau, dùng Thuật toán Ford-Bellman thực hiện các yêu cầu sau (2.5 điểm)  a) Xây dựng Ma trận trọng số (1.0 điểm) b) Tìm đường đi ngắn nhất từ một đỉnh $S \Rightarrow 1$ (S bắt đầu từ đỉnh 1) đến tất cả các đỉnh còn lại của đồ thị (1.5 điểm)	2.5
Tổng điểm		10.0đ

Đáp án (Tự Luận)

Câu 1:

- Nguyên lý:

- ✓ Bắt đầu tìm kiếm từ một đỉnh v nào đó của đồ thị.
- ✓ Sau đó chọn u là một đỉnh tùy ý kề với v (với đồ thị có hướng thì u là đỉnh sau, v là đỉnh đầu của cung uv)

- ✓ Lập lại quá trình này với u cho đến khi không tìm được đỉnh kế tiếp theo nữa thì trở về đỉnh ngay trước đỉnh mà không thể đi tiếp để tìm qua nhánh khác

- Cài đặt thuật toán DFS bằng Độ quy

B1: Lấy s là một đỉnh của đồ thị

B2: Đặt $v = s$

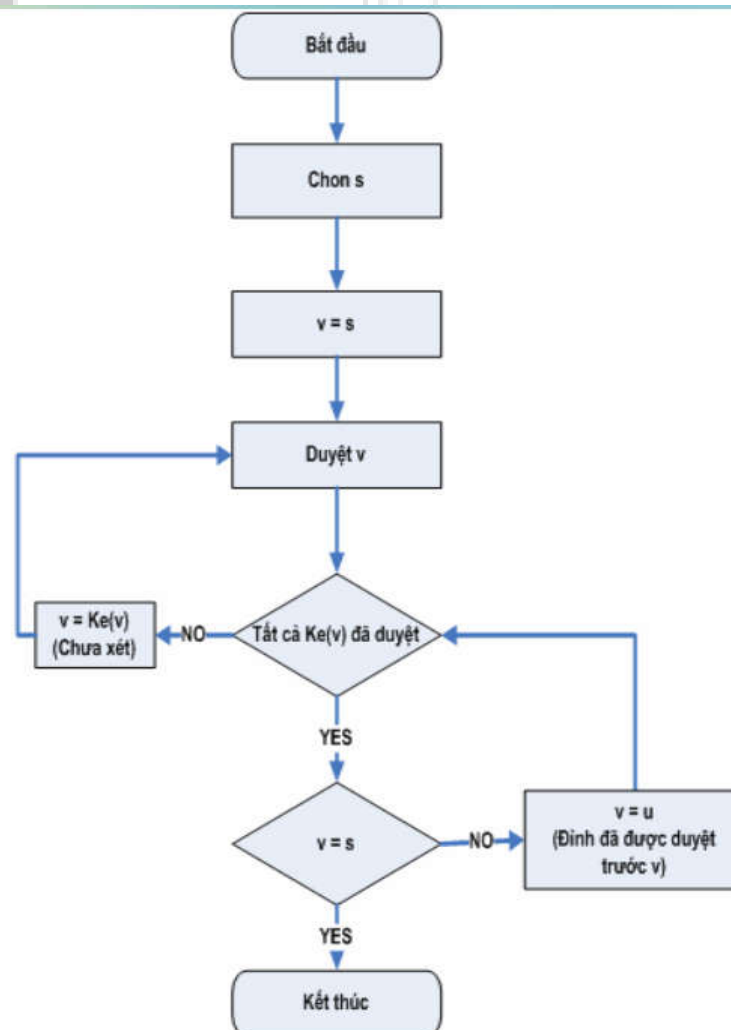
B3: Duyệt đỉnh v

B4: Nếu $\forall$ đỉnh k của v đều được duyệt, đặt $v =$ đỉnh đã được duyệt trước đỉnh v , Nếu $v = s$ thì đi đến Bước 6, ngược lại trở lại Bước 3.

B5: Chọn u là đỉnh k chưa được duyệt của v , đặt $v = u$, trở lại Bước 3

B6: Kết thúc

- Lưu đồ thuật toán



Câu 2:

```
def dfs_recursive(graph, v, visited):
    visited[v] = True
    print(v, end=' ')
    for neighbor in graph[v]:
        if not visited[neighbor]:
            dfs_recursive(graph, neighbor, visited)
```

Câu 3:

a. Ma trận trọng số

	1	2	3	4	5
1	∞	1	∞	∞	3
2	∞	∞	3	3	8
3	∞	∞	∞	1	-5
4	∞	∞	2	∞	∞
5	∞	∞	∞	4	∞

b. Đường đi ngắn nhất từ một đỉnh $S \Rightarrow 1$ (S bắt đầu từ đỉnh 1) đến tất cả các đỉnh còn lại của đồ thị:

k	d[5], Truoc[5]	d[4], Truoc[4]	d[3], Truoc[3]	d[2], Truoc[2]
1	3, 1	∞ , 1	∞ , 1	1, 1
2	3, 1	4, 2	4, 2	1, 1
3	-1, 3	4, 2	4, 2	1, 1
4	-1, 3	3, 5	4, 2	1, 1
5	-1, 3	3, 5	4, 2	1, 1