

BỘ XÂY DỰNG
TRƯỜNG ĐHXD MIỀN TÂY

ĐÁP ÁN – THANG ĐIỂM
KỲ THI KTHP HỌC KỲ I NĂM HỌC 2023-2024

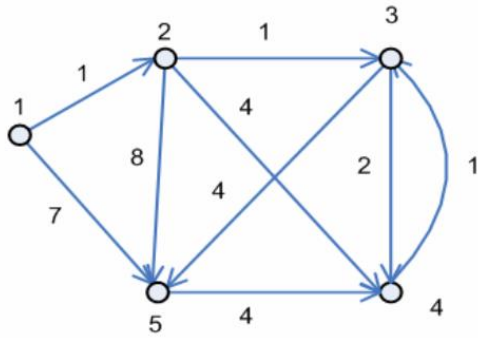
Trình độ: ĐẠI HỌC; Ngày thi: 05/01/2024

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	C	Tập hợp các đỉnh và cạnh	0.25
2	C	Một đồ thị có hướng không thể đi từ một đỉnh này đến mọi đỉnh khác	0.25
3	B	20	0.25
4	D	Đồ thị cây	0.25
5	A	G1 và G2 không đẳng cấu với nhau.	0.25
6	D	Bài toán TSP	0.25
7	A	33	0.25
8	A	Mọi đồ thị đều có thể tô được với chỉ 4 màu.	0.25
9	D	Đường đi qua tất cả các đỉnh một lần và quay về đỉnh xuất phát	0.25
10	C	Cả A và B	0.25
11	C	Hamilton	0.25
12	B	$(N - 1)$ Cạnh	0.25
13	C	Được kết nối tối thiểu	0.25
14	A	Thuật toán Floyd	0.25
15	A	$(m.i + 1)$ đỉnh và $(m.i)$ cạnh.	0.25
16	B	3 màu	0.25
17	B	In-order	0.25
18	A	$n(n-1)/2$	0.25
19	B	10	0.25
20	C	A: red, B: green, C: green, D: blue, E: yellow	0.25
II		TỰ LUẬN	5.0

1	Câu 1	Trình bày Thuật toán Ford-Bellman để tìm đường đi ngắn nhất trên đồ thị liên thông (Lưu ý: Input, Output, Mô tả thuật toán bằng ngôn ngữ giả, đánh giá độ phức tạp của thuật toán theo tiêu chí thời gian thực thi)	1.5
2	Câu 2	Dùng ngôn ngữ lập trình Python để hiện thực Thuật toán Dijkstra	1.0
3	Câu 3	Cho đồ thị như hình sau, dùng Thuật toán Dijkstra thực hiện các yêu cầu sau (2.5 điểm)  a) Xây dựng ma trận kề (1 điểm) b) Tìm đường đi ngắn nhất (1.5 điểm)	2.5
		Tổng điểm	10,0đ

Đáp án (Tự Luận)

1. Câu 1:

■ Đầu vào:

- Đồ thị có hướng $G=(V,E)$ với n đỉnh.
- $s \in V$ là đỉnh xuất phát.
- $a[u,v]$, $u,v \in V$ là ma trận trọng số

■ Đầu ra :

- Khoảng cách từ s đến tất cả các đỉnh còn lại $d[v]$, $v \in V$.
- $Truoc[v]$, $v \in V$ là đỉnh đi trước v trong đường đi ngắn nhất từ s đến v

```

void Ford_Bellman()
{
    for (v ∈ V) /* Khởi tạo d và Truoc */
    {
        d[v] = a[s,v];
        Truoc[v] = s;
    }
    d[s] = 0;
    for (k = 1; k < n-1; k++)
        for (v ∈ V \ {s})
            for (u ∈ V)
                if (d[v] > d[u] + a[u,v] )
                {
                    d[v] = d[u] + a[u,v] ;
                    Truoc[v] = u;
                }
    } /* Độ phức tạp của thuật toán là O(n3) */

```

2. Câu 2:

```

class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                       for row in range(vertices)]
    def printSolution(self, dist):
        print("Vertex \t Distance from Source")
        for node in range(self.V):
            print(node, "\t\t", dist[node])
    def minDistance(self, dist, sptSet):
        min = 1e7
        for v in range(self.V):
            if dist[v] < min and sptSet[v] == False:
                min = dist[v]
                min_index = v
        return min_index

    def dijkstra(self, src):
        dist = [1e7] * self.V
        dist[src] = 0
        sptSet = [False] * self.V

        for cout in range(self.V):
            u = self.minDistance(dist, sptSet)
            sptSet[u] = True
            for v in range(self.V):
                if (self.graph[u][v] > 0 and
                    sptSet[v] == False and
                    dist[v] > dist[u] + self.graph[u][v]):
                    dist[v] = dist[u] + self.graph[u][v]
        self.printSolution(dist)

```

3. Câu 3:

a. Ma trận kề

	1	2	3	4	5
1	∞	1	∞	∞	7
2	∞	∞	1	4	8
3	∞	∞	∞	2	4
4	∞	∞	1	∞	∞
5	∞	∞	∞	4	∞

b. Đường đi ngắn nhất

T	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5
2, 3, 4, 5	1, 1	$\infty, 1$	$\infty, 1$	7, 1
3, 4, 5		2, 2	5, 2	7, 1
4, 5			4, 3	6, 3
E				6, 3
$\emptyset$	1, 1	2, 2	4, 3	6, 3