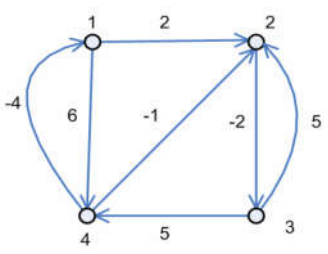


ĐÁP ÁN ĐỀ THI CHÍNH THỨC

(Đáp án - thang điểm gồm 04 trang)

Câu	Đáp án	Nội dung	Thang điểm
I		TRẮC NGHIỆM	5.0
1	C	Travelling salesman problem	0.25
2	C	G không có đường đi Euler	0.25
3	B	20	0.25
4	D	Bài toán Königsberg	0.25
5	A	G_1 và G_2 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	Tất cả câu trên đều đúng	0.25
10	C	1	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	Đồ thị chính quy	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 Dijkstra 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	<p>Cho đồ thị như hình sau, dùng Thuật toán Floyd thực hiện các yêu cầu sau (2.5 điểm)</p>  <p>a) Xây dựng ma trận Đường đi – D (1 điểm) b) Xây dựng ma trận Lưu vết - P (1 điểm) c) Tìm đường đi ngắn nhất từ đỉnh 1 -> 3 (0.5 điểm)</p>	2.5
Tổng điểm			10,0đ

Đáp án (Tự Luận)

1. Câu 1:

- ❖ Thuật toán Dijkstra dùng để tìm đường đi ngắn nhất từ đỉnh s đến các đỉnh còn lại trong đồ thị.
- ❖ Được sử dụng cho đồ thị không có cung trọng số âm.
- ❖ Thuật toán
 - 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 $d[v]$, $v \in V$.
 - $Truc[v]$, $v \in V$ là đỉnh đi trước v trong đường đi ngắn nhất từ s đến v .

```

void Dijkstra;{
  for (v ∈ V) /* Khởi tạo d và Truoc */ {
    d[v] = a[s,v];
    Truoc[v] = s;
  }
  d[s] = 0; T = V \ {s};
  while (T != ∅) {
    Tìm u ∈ T sao cho d(u) = min { d(z): z ∈ T }
    T = T \ {u}; /* Cố định nhãn của u */
    for (v ∈ T) do
      if (d[v] > d[u] + a[u,v] ) then
      {
        d[v] = d[u] + a[u,v] ;
        Truoc[v] = u;
      }
    }
  }
} /* Độ phức tạp của thuật toán là O(n2) */

```

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:

- Khởi tạo hai ma trận:
D: Ma trận lưu giá trị đường đi.
P: Ma trận lưu vết.

$$D0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 6 \\ 0 & 0 & -2 & 0 \\ 0 & 5 & 0 & 5 \\ -4 & -1 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$P0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & 3 \\ 4 & 4 & 0 & 0 \end{bmatrix} \end{matrix}$$

- $k=3$

$$D3 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 5 \\ 0 & 0 & -2 & 3 \\ 0 & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \end{matrix}$$

$$P3 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 3 \\ 0 & 3 & 0 & 3 \\ 4 & 1 & 2 & 0 \end{bmatrix} \end{matrix}$$

Kết quả các bước lặp của thuật toán:

- $k=1$

$$D1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 6 \\ 0 & 0 & -2 & 0 \\ 0 & 5 & 0 & 5 \\ -4 & -2 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$P1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & 3 \\ 4 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

- $k=4$

$$D4 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 5 \\ -1 & 0 & -2 & 3 \\ 1 & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \end{matrix}$$

$$P4 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 0 & 2 & 3 \\ 4 & 1 & 0 & 3 \\ 4 & 1 & 2 & 0 \end{bmatrix} \end{matrix}$$

- $k=2$

$$D2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 6 \\ 0 & 0 & -2 & 0 \\ 0 & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \end{matrix}$$

$$P2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & 3 \\ 4 & 1 & 2 & 0 \end{bmatrix} \end{matrix}$$

Vậy đường đi ngắn nhất từ
đỉnh 1 đến đỉnh 3 là: $1 \rightarrow 2 \rightarrow 3$.
Với trọng số = 0.