



ใบงานที่ 16

เรื่อง โครงสร้างข้อมูลกราฟ

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย สารินทร์ อินดีะรักษา รหัส 65543206082-1

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- สร้างโค้ดโปรแกรมตามตัวอย่างในเอกสารประกอบการสอนสัปดาห์ที่7
- แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

```
char Head[MaxNode][Block] = {
    {'A', '-', '1', '1'},
    {'B', '-', '3', '2'},
    {'C', '-', '2', '5'},
    {'D', '-', '0', NULL},
}; //Declare array and keep Head Node of graph
char Edge[MaxEdge][2] = {
    {'2', 'B'},
    {'5', 'A'},
    {'9', 'C'},
    {'7', 'D'},
    {'8', 'B'},
    {'6', 'D'},
}; //Declare array and keep Edge of graph
```

- Head[MaxNode][Block]:
 - MaxNode และ Block คือค่าคงที่หรือตัวแปรที่กำหนดขนาดของอาเรย์ Head ในมิติแนวนอนและแนวตั้งตามลำดับ
 - อาเรย์ Head ใช้เก็บข้อมูลของโหนด (Node) ในกราฟ โดยแต่ละแถวของอาเรย์จะเก็บข้อมูลของโหนดหนึ่งโหนด โดยมีรายละเอียดดังนี้
 - คอลัมน์แรกเก็บตัวอักษรที่แทนชื่อของโหนด
 - คอลัมน์ที่สองเก็บตัวอักษร '-' ใช้เพื่อแสดงความสัมพันธ์ระหว่างโหนดในกราฟ
 - คอลัมน์ที่สามเก็บตัวเลขของโหนด
 - คอลัมน์ที่สี่เก็บค่า NULL
- Edge[MaxEdge][2]:
 - MaxEdge คือค่าคงที่หรือตัวแปรที่กำหนดขนาดของอาเรย์ Edge ในมิติแนวนอน
 - อาเรย์ Edge ใช้เก็บข้อมูลเกี่ยวกับเส้นเชื่อม (Edge) ในกราฟ แต่ละแถวของอาเรย์จะเก็บข้อมูลของเส้นเชื่อมหนึ่งเส้น โดยมีรายละเอียดดังนี้
 - คอลัมน์แรกเก็บตัวเลขที่แทนค่าความหมายของเส้นเชื่อม
 - คอลัมน์ที่สองเก็บตัวอักษรที่แทนชื่อของโหนดที่เส้นเชื่อมไปยัง

```

void DispHead() //Display Head in 2D Array
{
    int i,j; //i=Row, j=Column
    printf("NODE...\n");
    printf("No. Name Data Edge Pointer\n");
    for (i=0;i<MaxNode;i++) //row loop
    {
        printf("%d ",i+1); //Display number of Row
        for (j=0;j<Block;j++) //column loop
            printf("%c ",Head[i][j]); //Display Node
        printf("\n");
    }
}

void DispEdge() //Display Edge in 2D Array
{
    int i,j; //i=Row, j=Column
    printf("EDGE...\n");
    printf("No. Weight Node\n");
    for (i=0;i<MaxEdge;i++) //row loop
    {
        printf("%d ",i+1); //Display number of Row
        for (j=0;j<2;j++) //column loop
            printf("%c ",Edge[i][j]); //Display Node
        printf("\n");
    }
}

```

- void DispHead():

- ฟังก์ชันนี้ใช้สำหรับแสดงข้อมูลที่เก็บในอาร์เรย์ Head ซึ่งเก็บข้อมูลของโหนด (Nodes) ในกราฟ.
- ฟังก์ชันจะแสดงข้อมูลโหนดทุกโหนดทีละโหนดโดยใช้ลูปซ้อนลูป ในการวนลูปเพื่อเข้าถึงทุกแถวและคอลัมน์ในอาร์เรย์ Head.

- ข้อมูลที่แสดงประกอบด้วย

- หมายเลขของแถว (Row number)
- ชื่อของโหนด (Node name)
- ข้อมูลเพิ่มเติมของโหนด (Node data)
- ตัวแทนของความสัมพันธ์ระหว่างโหนด (Edge pointer)

- void DispEdge():

- ฟังก์ชันนี้ใช้สำหรับแสดงข้อมูลที่เก็บในอาร์เรย์ Edge ซึ่งเก็บข้อมูลเส้นเชื่อม (Edges) ในกราฟ.
- ฟังก์ชันจะแสดงข้อมูลเส้นเชื่อมทุกเส้นทีละเส้นโดยใช้ลูปซ้อนลูป ในการวนลูปเพื่อเข้าถึงทุกแถวและคอลัมน์ในอาร์เรย์ Edge.

- ข้อมูลที่แสดงประกอบด้วย:

- หมายเลขของแถว (Row number)
- น้ำหนักของเส้นเชื่อม (Edge weight)
- ชื่อของโหนดปลายทางของเส้นเชื่อม (Node name)

```

void DispSetOfVertex() //Display set of Vertex
{
    int i;
    printf("\nSet of Vertex = {");
    for (i=0;i<MaxNode;i++)
    {
        printf("%c",Head[i][0]); //Display each node name
        if(i != MaxNode-1)
            printf(",");
    }
    printf("}\n");
}

void DispSetOfEdge() //Display set of Edge
{
    int i,j,AmtEdge,PT;
    printf("\nSet of Edge = {");
    for (i=0;i<MaxNode;i++) //row loop
    {
        AmtEdge=Head[i][2]-48; //Convert Character to Integer for Amount of Edge
        PT=Head[i][3]-48; //Convert Character to Integer for Start point of Array
        for (j=0;j<AmtEdge;j++) //Loop follow by Amount of Edge
        {
            printf("(%c%c)%c,",Head[i][0],Edge[PT-1+j][1],Edge[PT-1+j][0]);
        }
        //Show Edge and Weight
    }
    printf("}\n");
}

```

- void DispSetOfVertex():

- ฟังก์ชันนี้ใช้สำหรับแสดงเซตของ Vertex (โหนด) ในกราฟ
- ฟังก์ชันจะวนลูปผ่านทุกโหนดในอาร์เรย์ Head และแสดงชื่อของแต่ละโหนด void

DispSetOfEdge():

- ฟังก์ชันนี้ใช้สำหรับแสดงเซตของ Edge (เส้นเชื่อม) ในกราฟ
- ฟังก์ชันจะวนลูปผ่านทุกโหนดในอาร์เรย์ Head และสำหรับแต่ละโหนดจะดูค่าจำนวนเส้นเชื่อม (Amount of Edge) และจุดเริ่มต้นของอาร์เรย์ Edge (Start point of Array) ที่เก็บไว้ใน Head
- จากนั้นฟังก์ชันจะวนลูปผ่านเส้นเชื่อมที่เกี่ยวข้องกับโหนดนั้น และแสดงข้อมูลเส้นเชื่อมและน้ำหนักของเส้นเชื่อม

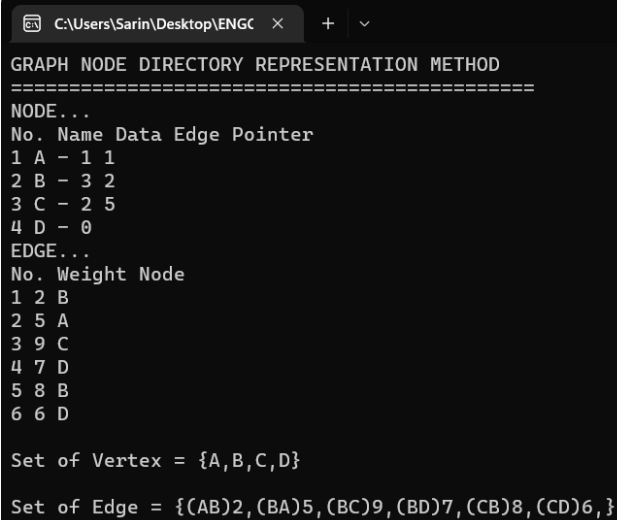
```

int main()
{
    printf("GRAPH NODE DIRECTORY REPRESENTATION METHOD\n");
    printf("=====\\n");
    DispHead();
    DispEdge();
    DispSetOfVertex();
    DispSetOfEdge();
    getch();
    return(0);
} //End Main

```

- แสดงข้อความ "GRAPH NODE DIRECTORY REPRESENTATION METHOD" บนหน้าจอ
- เรียกใช้ฟังก์ชัน DispHead() เพื่อแสดงข้อมูลที่เก็บในอาร์เรย์ Head
- เรียกใช้ฟังก์ชัน DispEdge() เพื่อแสดงข้อมูลที่เก็บในอาร์เรย์ Edge
- เรียกใช้ฟังก์ชัน DispSetOfVertex() เพื่อแสดงเซตของโหนดในกราฟ.
- เรียกใช้ฟังก์ชัน DispSetOfEdge() เพื่อแสดงเซตของเส้นเชื่อมในกราฟ
- Getch : รอรับอินพุตจากผู้ใช้ก่อนที่โปรแกรมจะจบการทำงาน
- return(0) : ส่งค่า 0 กลับเป็นผลลัพธ์ของฟังก์ชัน main() เพื่อแสดงว่าโปรแกรมทำงานสมบูรณ์

ผลลัพธ์การทดลอง



```

C:\Users\Sarin\Desktop\ENG C >
GRAPH NODE DIRECTORY REPRESENTATION METHOD
=====\\n
NODE...
No. Name Data Edge Pointer
1 A - 1 1
2 B - 3 2
3 C - 2 5
4 D - 0
EDGE...
No. Weight Node
1 2 B
2 5 A
3 9 C
4 7 D
5 8 B
6 6 D

Set of Vertex = {A,B,C,D}
Set of Edge = {(AB)2,(BA)5,(BC)9,(BD)7,(CB)8,(CD)6,}

```

สรุปผลการทดลอง

การทดลองนี้แสดงข้อมูลของกราฟโดยใช้โครงสร้างข้อมูลแบบ Node directory method ซึ่งเป็นการแสดงข้อมูลกราฟที่มีโหนดและเส้นเชื่อมบนหน้าจอในรูปแบบตารางและเซต Vertex และ Edge ของกราฟที่ถูกสร้างขึ้น

สื่อ / เอกสารอ้างอิง

ไฟล์ประกอบการสอนสัปดาห์ที่ 8 ของ อาจารย์ ปิยพล ยืนยงสถาวร เรื่อง : โครงสร้างข้อมูลกราฟ