

**Tugas :**

Buatlah program tentang algoritma Algoritma Breadth First Search (BFS) dalam bahasa C. Lalu, jelaskan terkait algoritma tersebut dan bagaimana prinsip queue digunakan dalam algoritma tersebut!

**Source code :**

```
/*
    Nama File   : algoritmaBFS.c
    Nama        : Manja Fani Oktavia
    Nim         : 22343056
*/
#include <stdio.h>
#include <stdlib.h>

#define MAX_QUEUE_SIZE 100
#define MAX_VERTEX_NUM 100

typedef struct {
    int data[MAX_QUEUE_SIZE];
    int front, rear;
} Queue;

void initQueue(Queue *q) {
    q->front = -1;
    q->rear = -1;
}

int isEmpty(Queue *q) {
    return q->front == -1;
}

int isFull(Queue *q) {
    return q->rear == MAX_QUEUE_SIZE - 1;
}

void enqueue(Queue *q, int value) {
    if (isFull(q)) {
        printf("Queue overflow\n");
        exit(1);
    }
    if (isEmpty(q)) {
        q->front = 0;
    }
    q->rear++;
    q->data[q->rear] = value;
}
```

```
}
```

```
int dequeue(Queue *q) {  
    if (isEmpty(q)) {  
        printf("Queue underflow\n");  
        exit(1);  
    }  
    int value = q->data[q->front];  
    if (q->front == q->rear) {  
        q->front = -1;  
        q->rear = -1;  
    } else {  
        q->front++;  
    }  
    return value;  
}
```

```
void bfs(int adjMatrix[MAX_VERTEX_NUM][MAX_VERTEX_NUM], int  
visited[MAX_VERTEX_NUM], int start, int numVertices) {  
    Queue q;  
    initQueue(&q);  
    visited[start] = 1;  
    printf("%d ", start);  
    enqueue(&q, start);  
    while (!isEmpty(&q)) {  
        int curr = dequeue(&q);  
        for (int i = 0; i < numVertices; i++) {  
            if (adjMatrix[curr][i] == 1 && !visited[i]) {  
                visited[i] = 1;  
                printf("%d ", i);  
                enqueue(&q, i);  
            }  
        }  
    }  
}
```

```
int main() {  
    int adjMatrix[MAX_VERTEX_NUM][MAX_VERTEX_NUM],  
    visited[MAX_VERTEX_NUM];  
    int numVertices, start;  
    printf("Programmer : Manja Fani Okavia\n NIM : 22343056\n");  
    printf("Enter the number of vertices: ");  
    scanf("%d", &numVertices);  
    printf("Enter the adjacency matrix: ");  
    for (int i = 0; i < numVertices; i++) {  
        for (int j = 0; j < numVertices; j++) {
```

```

        scanf("%d", &adjMatrix[i][j]);
    }
}
printf("Enter the starting vertex: ");
scanf("%d", &start);
for (int i = 0; i < numVertices; i++) {
    visited[i] = 0;
}
printf("BFS traversal: ");
bfs(adjMatrix, visited, start, numVertices);
printf("\n");
return 0;
}

```

### Hasil :

```

C:\Users\user\Downloads\algoritmaBFS.c [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project: C:\Users\user\Downloads\algoritmaBFS.c
Compiler: TDM-GCC 9.2.0 64-bit Release
55 } return value;
56 }
57
58 void bfs(int adjMatrix, int numVertices, int start, int visited[]) {
59     Queue q;
60     initQueue(&q);
61     visited[start] = 1;
62     printf("%d ", start);
63     enqueue(&q, start);
64     while (!isEmpty(q)) {
65         int curr = dequeue(&q);
66         for (int i = 0; i < numVertices; i++) {
67             if (adjMatrix[curr][i] & visited[i] == 0) {
68                 visited[i] = 1;
69                 printf("%d ", i);
70                 enqueue(&q, i);
71             }
72         }
73     }
74 }
75
76 int main() {
77     int adjMatrix[10][10];
78     int numVertices;
79     printf("Programmer : Manja Fani Okavia\n");
80     printf("NIM : 22343056\n");
81     scanf("%d", &numVertices);
82     printf("Enter the number of vertices: ");
83     scanf("%d", &numVertices);
84     printf("Enter the adjacency matrix: ");
85     for (int i = 0; i < numVertices; i++) {
86         for (int j = 0; j < numVertices; j++) {
87             scanf("%d", &adjMatrix[i][j]);
88         }
89     }
90     printf("Enter the starting vertex: ");
91     scanf("%d", &start);
92     bfs(adjMatrix, numVertices, start, visited);
93     printf("\n");
94     return 0;
95 }

```

Programmer : Manja Fani Okavia  
 NIM : 22343056  
 Enter the number of vertices: 1  
 Enter the adjacency matrix: 2  
 Enter the starting vertex: 3  
 BFS traversal: 3  
 Process exited after 14.93 seconds with return value 0  
 Press any key to continue . . .

Compiler: TDM-GCC 9.2.0 64-bit Release  
 Errors: 0  
 Warnings: 0  
 Output File Name: algoritmaBFS.exe  
 Output Size: 324,768 bytes  
 Compilation Time: 0,36s

### Penjelasan singkat tentang algoritma BFS:

Algoritma BFS adalah salah satu algoritma penelusuran grafik (graph traversal) yang digunakan untuk mengunjungi setiap simpul pada grafik atau pohon secara sistematis, dengan mengunjungi semua simpul yang terhubung pada level yang sama sebelum melanjutkan ke simpul yang terhubung pada level berikutnya. Algoritma ini mengimplementasikan konsep pengunjungan terlebih dahulu pada level yang lebih rendah sebelum pengunjungan pada level yang lebih tinggi (level-order traversal).

**Prinsip queue digunakan dalam algoritma BFS** untuk memastikan bahwa simpul yang ditelusuri pada level yang sama diproses secara urut (FIFO - First In First Out). Pada setiap langkah BFS, simpul ditambahkan ke dalam antrian atau queue dan setelah pengunjungan simpul tersebut, simpul tersebut dikeluarkan dari queue dan simpul terhubung pada level

berikutnya dimasukkan ke dalam queue. Dalam implementasi program di atas, queue diwakili oleh array `queue[]`, `front` dan `rear` digunakan sebagai indeks yang menunjukkan posisi pertama dan terakhir pada queue, serta queue diinisialisasi dengan nilai -1.