

Alvin Anugerah Pratama
22343019
Informatika NK

PROGRAM TENTANG ALGORITMA BREADTH FIRST SEARCH (BFS)

```
1. // Created by 22343019_Alvin Anugerah Pratama
2. #include<stdio.h>
3. #include<stdlib.h>
4.
5. #define MAX 100
6.
7. int adj[MAX][MAX];
8. int visited[MAX];
9. int queue[MAX];
10.     int front = -1, rear = -1;
11.
12.     void bfs(int start, int n) {
13.         int i, v;
14.         visited[start] = 1;
15.         queue[++rear] = start;
16.
17.         while(front != rear) {
18.             v = queue[++front];
19.             for(i=0; i<n; i++) {
20.                 if(adj[v][i] == 1 && visited[i] == 0) {
21.                     visited[i] = 1;
22.                     queue[++rear] = i;
23.                 }
24.             }
25.             printf("%d ", v);
26.         }
27.     }
28.
29.     int main() {
30.         int i, j, n, start;
31.         printf("Masukkan jumlah simpul: ");
32.         scanf("%d", &n);
33.         printf("Masukkan matriks ketetanggaan:\n");
34.         for(i=0; i<n; i++) {
35.             for(j=0; j<n; j++) {
36.                 scanf("%d", &adj[i][j]);
37.             }
38.         }
39.         printf("Masukkan simpul awal: ");
40.         scanf("%d", &start);
41.         printf("Jalur BFS: ");
42.         bfs(start, n);
43.         printf("\n");
44.         return 0;
45.     }
```

Algoritma Breadth First Search (BFS) adalah algoritma pencarian graf yang melintasi semua simpul dari simpul awal (start node) dengan mengunjungi semua simpul yang terhubung secara berurutan terlebih dahulu sebelum mengeksplorasi simpul-simpul yang terhubung secara tidak langsung. Algoritma BFS menggunakan struktur antrian untuk menerapkan prinsip “First In First Out” (FIFO), sehingga node yang dikunjungi selalu berada dalam antrian sebelum node yang tidak dikunjungi.

Pada program di atas, setiap simpul dari graf direpresentasikan dalam bentuk matriks ketetanggaan. Setiap elemen matriks ketetanggaan menunjukkan ada atau

tidaknya sisi yang menghubungkan dua simpul. Saat algoritma BFS berjalan, node asli ditambahkan ke antrian dan ditandai sebagai telah dikunjungi. Kemudian node dalam antrian dihapus satu per satu dan node yang terhubung langsung ke node ini dicari. Setiap node yang belum dikunjungi yang terhubung langsung ke node ini ditambahkan ke antrian dan ditandai sebagai telah dikunjungi. Proses ini diulang sampai semua simpul dari graf telah dikunjungi.

Dalam implementasinya, prinsip antrian digunakan untuk menyimpan node yang sudah dikunjungi dan belum dikunjungi. Setelah sebuah node dikunjungi, node tersebut diantrekan untuk diperiksa dalam rangkaian node yang terhubung. Ini dilakukan dengan terlebih dahulu mengunjungi setiap node pada level yang sama sebelum memeriksa node pada level yang lebih rendah. Selama proses BFS, node yang dikunjungi dihapus dari antrian dan node yang tidak dikunjungi ditambahkan ke antrian. Antrian FIFO mempertahankan urutan di mana node yang terhubung dieksplorasi secara berurutan, memastikan bahwa node pada level yang sama dikunjungi terlebih dahulu sebelum menjelajahi level yang lebih dalam.