

## **PROBLEM STATEMENT 1**

/\* Breadth First Search (BFS):

Application:Indexing web pages for search engines.

Example: A web crawler uses BFS to visit web pages systematically, starting from a seed URL and exploring links level by level. Nodes represent web pages. Edges represent hyperlinks. BFS ensures that pages at the same "depth" (distance from the starting page) are visited before moving to deeper levels. Write a program to simulate the indexing of web pages for a search engine using a Breadth-First Search (BFS) algorithm. \*/

```
#include <iostream>
#include <queue>    //to use queue in BFS
using namespace std;

class WebCrawler {
    int pages;           //number of web pages
    int links[10][10];   //adjacency matrix (edges)
    int visited[10];     //keep track of visited pages

public:
    //Function to read the web graph (pages and links)
    void readWeb() {
        int connections;
        cout<<"\nEnter number of web pages: ";
        cin>>pages;

        //initialize matrix and visited array to 0
        for (int i = 1; i <= pages; i++) {
            for (int j = 1; j <= pages; j++) {
                links[i][j] = 0;
            }
            visited[i] = 0;
        }

        cout << "Enter number of hyperlinks between pages: ";
        cin >> connections;

        cout << "Enter hyperlinks (from page to page):\n";
        for (int k = 0; k < connections; k++) {
            int u, v;
            cin >> u >> v;
            links[u][v] = 1;
            links[v][u] = 1;    //assuming links are two-way for this example
        }
    }
}
```

```

//BFS function to simulate crawling
void crawlBFS(int start) {
    queue<int> q;          //queue for BFS
    visited[start] = 1;      //mark the starting page as visited
    q.push(start);          //add starting page to the queue

    cout<<"BFS-based Web Crawling Start..." << endl;

    while (!q.empty()) {
        int current = q.front(); //get front page from queue
        q.pop();                //remove it from queue

        cout<<"Indexed Page "<<current<<endl;

        //explore all linked pages of the current page
        for (int i = 1; i <= pages; i++) {
            //if there is a link and the page is not visited
            if (links[current][i] == 1 && visited[i] == 0) {
                visited[i] = 1;      //mark as visited
                q.push(i);          //add page to queue
            }
        }
        cout<<"\nAll reachable web pages have been indexed!" << endl;
    }

    //function to start BFS crawling
    void startCrawling() {
        int start;
        cout << "Enter the starting web page number: ";
        cin >> start;
        crawlBFS(start);
    }
};

int main() {
    WebCrawler w;
    cout<<"Name:Manasvi Lunawat PRN:B24CE1136"<<endl;
    w.readWeb();
    w.startCrawling();
    return 0;
}

```

**OUTPUT:**

```
Name:Manasvi Lunawat PRN:B24CE1136
```

```
Enter number of web pages: 5
```

```
Enter number of hyperlinks between pages: 6
```

```
Enter hyperlinks (from page to page):
```

```
1 2
```

```
1 3
```

```
2 4
```

```
2 5
```

```
3 4
```

```
4 5
```

```
Enter the starting web page number: 1
```

```
BFS-based Web Crawling Start...
```

```
Indexed Page 1
```

```
Indexed Page 2
```

```
Indexed Page 3
```

```
Indexed Page 4
```

```
Indexed Page 5
```

```
All reachable web pages have been indexed!
```