|  |  |
| --- | --- |
| **Tree BFS** | |
| **int V = 5;**  **vi graph[V];**  **graph[0].push\_back(1);**  **graph[0].push\_back(2);**  **graph[1].push\_back(3);**  **graph[1].push\_back(4);**  **bool marked[V];**  **int level[V];**  **queue<int> q;**  **q.push(0);**  **level[0] = 0;**  **marked[0] = true;**  **while (!q.empty())**  **{**  **int x = q.front();**  **q.pop();**  **for (int i = 0; i < graph[x].size(); i++)**  **{**  **int b = graph[x][i];**  **if (!marked[b])**  **{**  **q.push(b);**  **level[b] = level[x] + 1;**  **marked[b] = true;**  **}**  **}**  **}**  **for (auto i = 0; i < V; i++)**  **{**  **cout << i << " " << level[i] << endl;**  **}** | **Output:-**  **0 0**  **1 1**  **2 1**  **3 2**  **4 2** |

**void bfs(vector<int> graph[], int v, int x)**

**{**

**int level[v];**

**bool marked[v];**

**queue<int> q;**

**q.push(x);**

**level[x] = 0;**

**marked[x] = true;**

**while (!q.empty())**

**{**

**x = q.front();**

**q.pop();**

**for (auto i = 0; i < graph[x].size(); i++)**

**{**

**int b = graph[x][i];**

**if (!marked[b])**

**{**

**q.push(b);**

**marked[b] = true;**

**level[b] = level[x] + 1;**

**}**

**}**

**}**

**for (auto i = 0; i < v; i++)**

**{**

**cout << i << " " << level[i] << endl;**

**}**

**}**