import java.util.\*;

public class BFS {

private List<List<Integer>> adjust;

public BFS(int vertices) {

adjust = new ArrayList<>();

for (int i = 0; i < vertices; i++) {

adjust.add(new ArrayList<>());

}

}

public void addEdge(int u, int v) {

adjust.get(u).add(v);

adjust.get(v).add(u);

}

public void bfs(int start) {

boolean[] visited = new boolean[adjust.size()];

LinkedList<Integer> queue = new LinkedList<>();

visited[start] = true;

queue.add(start);

while (!queue.isEmpty()) {

int node = queue.poll();

System.out.print(node + " ");

for (int neighbor : adjust.get(node)) {

if (!visited[neighbor]) {

visited[neighbor] = true;

queue.add(neighbor);

}

}

}

}

public static void main(String[] args) {

BFS graph = new BFS(5);

graph.addEdge(0, 1);

graph.addEdge(0, 2);

graph.addEdge(1, 3);

graph.addEdge(2, 4);

System.out.println("BFS starting from node 0:");

graph.bfs(0);

}

}