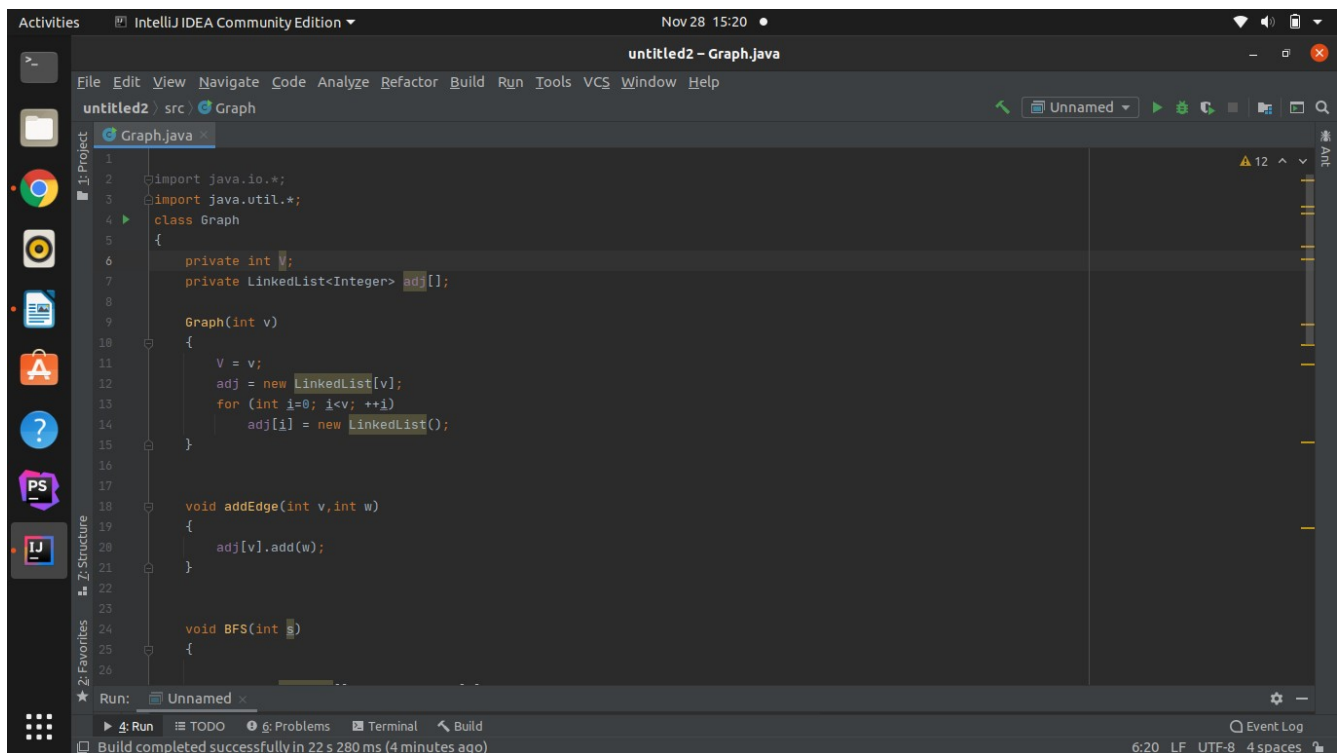


Breadth First Search

Defination:-

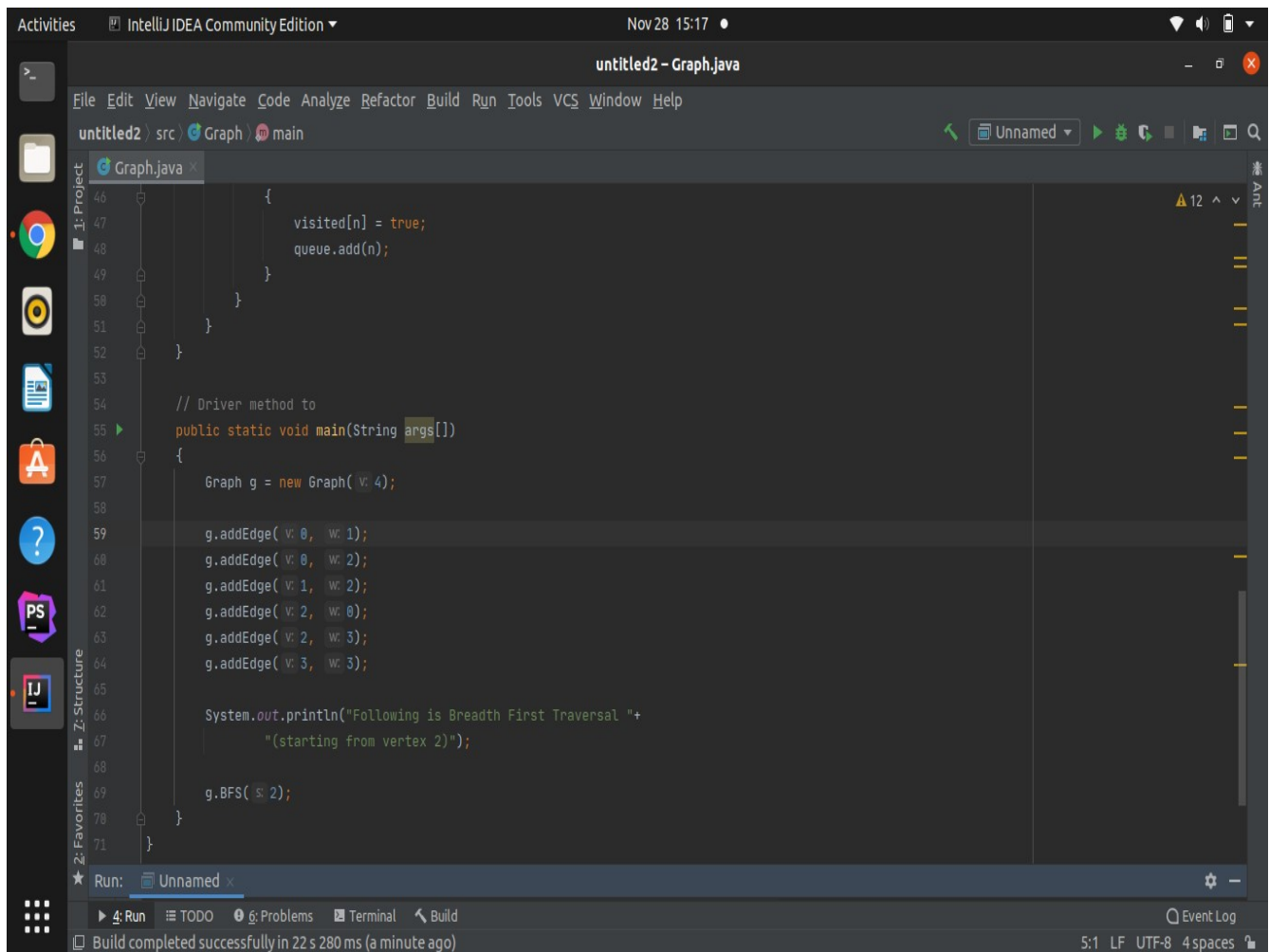
Breadth First Traversal (or Search) for a graph is similar to Breadth First Traversal of a tree .The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array. For simplicity, it is assumed that all vertices are reachable from the starting vertex.

Implementation



```
1  import java.io.*;
2
3  import java.util.*;
4
5  class Graph
6  {
7      private int V;
8      private LinkedList<Integer> adj[];
9
10     Graph(int v)
11     {
12         V = v;
13         adj = new LinkedList[V];
14         for (int i=0; i<V; ++i)
15             adj[i] = new LinkedList();
16     }
17
18     void addEdge(int v,int w)
19     {
20         adj[v].add(w);
21     }
22
23
24     void BFS(int s)
25     {
26     }
```

The screenshot shows the IntelliJ IDEA Community Edition interface. The main editor window displays the code for a Java class named `Graph`. The code includes imports for `java.io.*` and `java.util.*`, and defines a `Graph` class with a private integer `V` and an array of `LinkedList<Integer>` named `adj`. The `Graph(int v)` constructor initializes `V` and creates the `adj` array. The `addEdge(int v, int w)` method adds an edge between vertices `v` and `w`. The `BFS(int s)` method is also defined but its body is empty. The bottom status bar indicates that the build completed successfully in 22 s 280 ms (4 minutes ago).



Output:-

