Kunal Sachdeva

19115045

ADA LAB

BFS Traversal

```
#include<iostream>
#include <list>
using namespace std;
class Graph
{
	int V;
	list<int> *adj;
	public:
	Graph(int V)
	{
		this->V = V;
		adj = new list<int>[V];
	}
	void addEdge(int v, int w)
	{
		adj[v].push_back(w);
	}
	void BFS(int s)
	{
		bool *visited = new bool[V];
		for(int i = 0; i < V; i++)
			visited[i] = false;
		list<int> queue;
		visited[s] = true;
		queue.push_back(s);
		list<int>::iterator i;
		while(!queue.empty())
```

```
		{
			s = queue.front();
			cout << s << " ";
			queue.pop_front();
			for (i = adj[s].begin(); i != adj[s].end(); ++i)
			{
				if (!visited[*i])
				{
					visited[*i] = true;
					queue.push_back(*i);
				}
			}
		}
	}
};
int main()
{
	Graph g(4);
	g.addEdge(0, 1);
	g.addEdge(0, 2);
	g.addEdge(1, 2);
	g.addEdge(2, 0);
	g.addEdge(2, 3);
	g.addEdge(3, 3);
	cout << "Following is Breadth First Traversal (starting from vertex 2) \n";
	g.BFS(2);
	return 0;
}
```

Output:

Success #stdin #stdout 0s 5344KB comments (0)

stdin copy

```
Standard input is empty
```

stdout copy

```
Following is Breadth First Traversal (starting from vertex 2)
2 0 3 1
```