CS 61B: Lecture 29 Code

```
public class Graph {
  // Before calling dfs(), set every "visited" flag to false; O(|V|) time
  public void dfs(Vertex u) {
    u.visit();                              // Do some unspecified thing to u
    u.visited = true;                            // Mark the vertex u visited
    for (each vertex v such that (u, v) is an edge in E) {
      if (!v.visited) {
        dfs(v);
      }
    }
  }


  public void bfs(Vertex u) {
    for (each vertex v in V) {                                  // O(|V|) time
      v.visited = false;
    }
    u.visit(null);                          // Do some unspecified thing to u
    u.visited = true;                            // Mark the vertex u visited
    q = new Queue();                                           // New queue...
    q.enqueue(u);                                // ...initially containing u
    while (q is not empty) {
      v = q.dequeue();
      for (each vertex w such that (v, w) is an edge in E) {
        if (!w.visited) {
          w.visit(v);                       // Do some unspecified thing to w
          w.visited = true;                      // Mark the vertex w visited
          q.enqueue(w);
        }
      }
    }
  }
}


public class Vertex {
  protected Vertex parent;
  protected int depth;
  protected boolean visited;

  public void visit(Vertex origin) {
    this.parent = origin;
    if (origin == null) {
      this.depth = 0;
    } else {
      this.depth = origin.depth + 1;
    }
  }
}
```