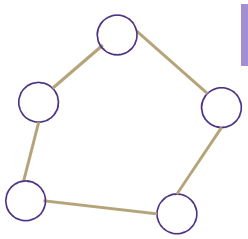
A detailed application

Bipartite (also called "2-colorable")

A graph is bipartite (also called 2-colorable) if the vertex set can be divided into two sets V_1, V_2 such that the only edges go between V_1 and V_2.

Called "2-colorable" because you can "color" V_1 red and V_2 blue, and no edge connects vertices of the same color.



If a graph contains an odd cycle, then it is not bipartite.

Try the example on the right, then proving the general theorem in the light purple box.

Help Robbie figure out how long to make the explanation
Pollev.com/robbie

BFS With Layers

```
search(graph)
    toVisit.enqueue(first vertex)
    mark first vertex as seen
    toVisit.enqueue(end-of-layer-marker)
    l=1
    while(toVisit is not empty)
        current = toVisit.dequeue()
        if(current == end-of-layer-marker)
            l++
            toVisit.enqueue(end-of-layer-marker)
        current.layer = l
        for (v : current.neighbors())
            if (v is not seen)
                mark v as seen
                toVisit.enqueue(v)
```

It's just BFS!

With some extra bells and whistles.

Lemma 3

If a graph has no odd-length cycles, then it is bipartite.

Wrapping it up

```
BipartiteCheck(graph) //assumes graph is connected!
   toVisit.enqueue(first vertex)
   mark first vertex as seen
   toVisit.enqueue(end-of-layer-marker)
   l="odd"
   while(toVisit is not empty)
      current = toVisit.dequeue()
        if(current == end-of-layer-marker)
              l++
           toVisit.enqueue(end-of-layer-marker)
      current.layer = l
      for (v : current.neighbors())
         if (v is not seen)
               mark v as seen
            toVisit.enqueue(v)
          else //v is seen
            if(v.layer == current.layer)
                 return "not bipartite" //intra-level edge
  return "bipartite" //no intra-level edges
```

On homework, you can tell us "assume BipartiteCheck was modified to handle disconnected graphs" if you want those handled automatically.
You just add a wrapper, like you've seen in 332.