James Audretsch

2/13/16

My partner for the project was Kristen. I will add her full name (this is her American name, not what is on the roster) by the time you see this I am sure

Time complexity analysis:

Starting from the beginning of the *flood* function, we have a **for loop** that scans through our *floodList* **n** times. *floodList.get(i)* occurs in **O(1)** time. We create 4 *Coords,* one for each direction. This is a constant time. We then call *change\_color* on all 4 *Coords.*

We check if the *Coord* exists and if the color is the same as our current *Coord.* This takes a constant time. Then, we use a function called *in\_floodList(next\_coord)* that searches through our *floodList* to see if the coordinate is already in it. This will take **O(n)** time. If it is not in the the *floodList*, then we add the new *Coord* to the list.

This last step is where we could have improved with a more efficient data structure, where instead of scanning through every item in our list to see if a block is already in it, we could have used a different structure to get a lower running time.

This gives us an approximate time complexity of: **O(n^2)**