Rafael Gomez
Stiven Agudelo

3.1
We utilized hashmap in order to complete the program. We used a buffered reader to read the file and save the the information that we are reading. In order to be able to save enough information to be able to do whatever the user asks for, we used a hashmap that is composed of another hashmap.Within the second hashmap we then put an arraylist in order to keep the successors together so they can be searched for after.
3.2
If there are 300,000 vertices then a matrix of that size will consume 90,000,000 spaces of memory. If you use a matrix for this particular problem you would need 670 GB of space.
3.3
We used a hashmap and pushed in a long with the id, so the number that we are starting with does not matter.
3.4
First we utilized a hashmap that has a linked list that is composed of integers, this is to keep track of the adjacent vertices. We then used another hashmap that was used to keep track of the colored used, the key is an integer that is the vertex and the value being another integer, however the integer represents a color.
3.5
The complexity of our program is worst case: O(n x 2m).
3.6
In the complexity n is the number of vertices and m the number of arcs.
4.2
0 -> [3,4]
1 -> [0,2,5]
2 -> [1,4,6]
3 -> [7]
4 ->[2]
5 ->
6 ->[2]
7 ->
4.3
Answer: b