Please enter your name and uID below.

Name:

uID:

Collaborators, if any, and how you collaborated:

Submission notes

- Due at 11:59 pm on Thursday, October 27th.
- Solutions must be typeset using one of the template files. For each problem, your answer must fit in the space provided (e.g. not spill onto the next page) *without* space-saving tricks like font/margin/line spacing changes.
- Upload a PDF version of your completed problem set to Gradescope.
- Teaching staff reserve the right to request original source/tex files during the grading process, so please retain these until an assignment has been returned.
- Please remember that for this problem set, you are allowed to collaborate in detail with your peers, as long as you cite them. However, you must write up your own solution, alone, from memory. If you do collaborate with other students in this way, you must identify the students and describe the nature of the collaboration. You are not allowed to create a group solution, and all work that you hand in must be written in your own words. Do not base your solution on any other written solution, regardless of the source.

1. (Bipartite Graphs, 40pts)

   A graph $G = (V, E)$ is *bipartite* if the vertices V can be partitioned into two subsets L and R, such that every edge has one vertex in L and the other in R.

   Prove that a connected graph G is bipartite if and only if every cycle in G has an even number of edges.[1]

   Hint: you prove an "if and only if" statement by proving both directions. So, (1) prove that if G is bipartite, then every cycle in G must have even length. Then (2) prove that if every cycle in G has even length, then G must be bipartite.

   To prove (2), start with some arbitrary vertex v and put it in R. If you put v's neighbors in L, and all of their neighbors in R, and so on, would you ever end up with an edge from L to L or R to R? Argue that this is couldn't happen if every cycle in G has an even number of edges.

[1] The *length* of a cycle is its number of edges. An even length cycle always has an odd number of vertices. For example, the walk $1, 2, 3, 4, 1$ uses the edges $12, 23, 34, 41$.

2. (Getting Gold, 25pts)

   Describe an algorithm to solve the PS5 programming problem by mapping the input to a graph and then using a variant of Whatever First Search (this task is similar to the solving of Flood Fill in Chapter 5.7). Your description must include the following:

   a. What are the vertices? What does each vertex represent? *For example, in Flood Fill, the vertices represent pixels in the input.*

   b. What are the edges and what do they represent (defining your edges carefully simplifies the problem)? *In Flood Fill, the edges are undirected and connect neighboring pixels of the same color.*

   c. If the vertices and/or edges have associated values, what are they? *In Flood Fill, vertices store the color of the associated pixel.*

   d. What problem do you need to solve on this graph? How does WFS solve this problem? *Solving Flood Fill is equivalent to turning* i, j*'s connected component to the provided color. WFS allows us to reach and mark all elements in* i, j*'s connected component.*

   e. What is the running time of your entire algorithm, including the construction of the graph, in terms of the input parameters W and H?