Question 1:

1.1

- From $https://en.wikipedia.org/wiki/Tree_(graph_theory)$:
  In graph theory, a tree is an undirected graph in which any two vertices are connected by exactly one path, or equivalently a connected acyclic undirected graph.

- So we are looking for a graph that: is acyclic and has only one path for any two vertices.

- For 'whether there is a cycle' part, use DFS, which gives the answer in $O(V + 2E) = O(n)$.

- For 'the connected vertices' part, this question is the same as: do we have $n - 1$ edges for n vertices? So we count the number of vertices:

- Go through the adjacency list to get the number of edges, if the number $= 2(n - 1)$ (n is the number of vertices and in the list every edge is counted twice so the number is doubled), which also takes $O(n)$ and also DFS tells that there is no cycle in graph, then we can say this graph is a tree.

1.2

- By using adjacency matrix, DFS would become $O(V^2) = O(n^2)$, and counting the edges also is the same, so total $O(n^2)$

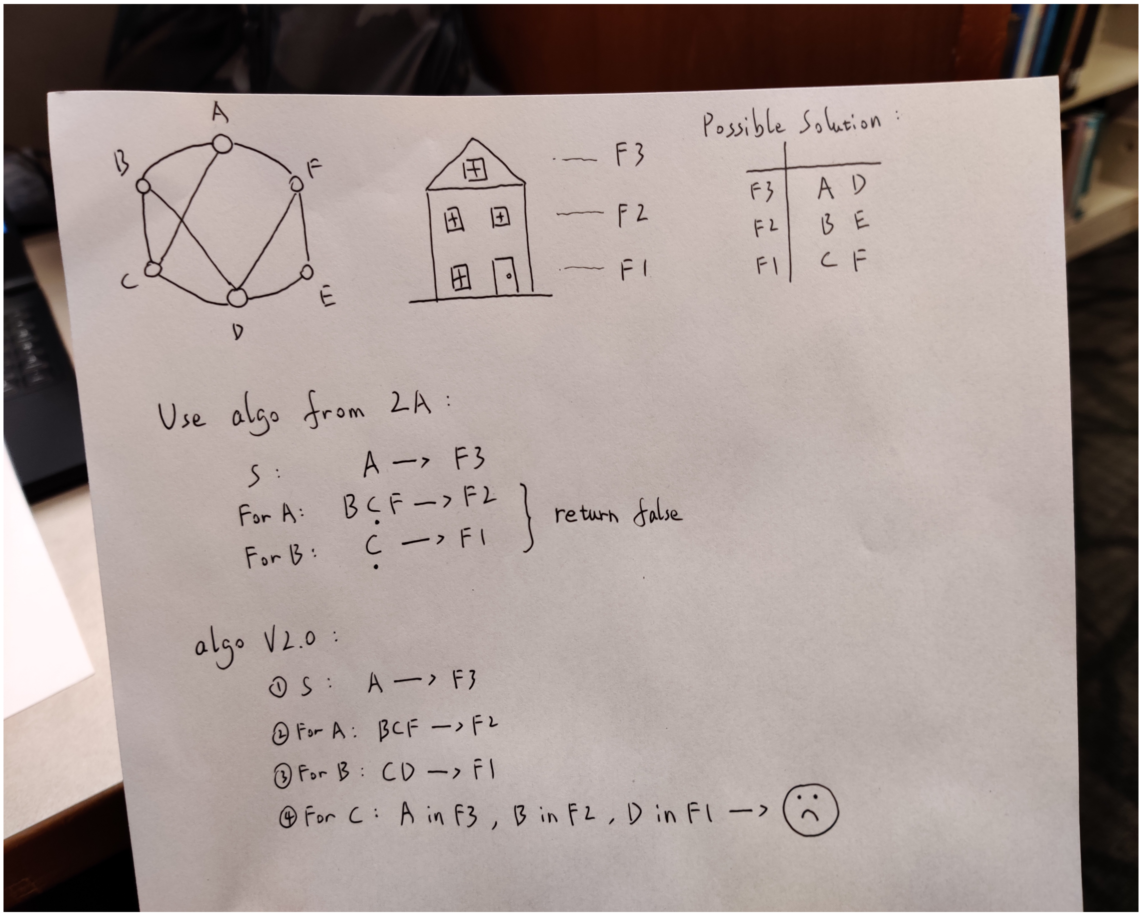
Question 2:

2.1

- Use an undirected graph to represent the relation, vertice: person, edge: hate each other. So the problem becomes: Can I seperate all vertices into two that no edge appears within any of the two groups.
- How to do that: Use BFS but look for update each time. Start from a random person S (this works since every one has at least someone he/she hates so starting point does not matter), put him/her to floor 1. look for the neighbors, put them into floor 2 and look for their neighbors and put them into floor 1, so on and so forth. This function stops when we want to put a person on one floor but find him/her been assigned to another floor already, then the function returns false and we can say this mission is impossible.
- Time: $O(V + E) = O(n + k)$

2.2

- It does not always work. And here is an example:



definitely not lax house

- As shown in the picture, these 6 persons can be put into a $f = 3$ house properly, however, if we use the original algorithm, this will happen:

- We put A into F3, then search for its neighbor and find BCF, we put them into F2 and continue search for neighbors of B and also find C, we need to put C into F1 and the function will tell us that since C has been reassigned, the mission is impossible. So the original one fails.

- Let me redesign it and add another rule: when we encounter a person that needs to be reassigned to another place, try doing so first and only stop the function when it is impossible to reassign and then return false.
- Algo v2.0:
- First we start with A: put A into F3
- Then we put A's neighbor into F2
- Then we put B's neighbor into F1
- Now it is turn for C's neighbor and we find that 3 neighbors of C: A is in F3, B is in F2 and D is in F1, no place to move, the function returns false. (However possible solution exists so this algo does not work in this example)

Question 3:

- For SSSP and no negative weight, use Dijkstra:
- Build two heaps for both virus (xheap) and antidote (yheap)
- For each vertice, we should record these info: its name, its shortest path to X: xdist, its shortest path to Y: ydist, and its status: neutral, infected, protected or damaged.
- Algorithm:
- Start with initializing all the vertices with $xdist = \infty$, $ydist = \infty$ and status: neutral. For X: xdist = 0, status: infected; Y: ydist = 0, status: protected
- Run Dijkstra
- The following function stops when either heap is empty or every vertice has its distance calculated:
- take the min elements from both xheap and yheap:
- If these two (xmin and ymin) have same name and have $xdist = ydist$, mark both as damaged
- If $xmin.xdist > ymin.ydist$, mark ymin as protected, and relax its neighbors in ydist, pop it from xheap and reinsert xmin into xheap
- If $xmin.xdist < ymin.ydist$, mark xmin as infected, and relax its neighbors in xdist, pop it from yheap and reinsert ymin into yheap
- Runtime: overall $O(ElogV)$, but compare to the original Dijkstra, we have to spend extra $O(1)$ time to update the info for vertices and O(logn) to rearrange heaps
- (with help from Ashley Wicks)