AGTA Tutorial 6

Please attempt the question before your tutorial.

1. Consider the following 2-player win-lose game on a graph. We are given a finite 2-player game graph $G = (V, E, v_0, F)$, where the vertices $V = V_1 \cup V_2$, are partitioned into the vertices V_1 belonging to player 1 and the vertices V_2 belonging to player 2. We are also given a start vertex $v_0 \in V$, and a set $F \subset V$ of "good" states. Starting from v_0, the player who owns the current state, v_i, chooses an neighbor $v_{i+1} \in V$ such that $(v_i, v_{i+1}) \in E$. This determines an (infinite) play, $\pi = v_0 v_1 v_2 \ldots$. The winner is determined as follows. Let $\inf(\pi)$ denote the set of vertices in V that occur infinitely often along the play π. The play π is a win for player 1 if $\inf(\pi) \cap F \neq \emptyset$, and otherwise it is loss for player 1 (i.e., a win for player 2).

   In other words, player 1's goal is to force the play to visit some vertex in F infinitely often, whereas player 2's goal is to make sure this does not happen. These games are in fact memorylessly determined.

   Describe an efficient (polynomial time, and as efficient as you can get it) algorithm for determining the winning player (i.e., the one with a winning strategy), and for computing a memoryless winning strategy for that player, given such a game.

   (Hint: repeatedly use the algorithm given in class for determining the winner in a game on a graph where the goal of player 1 is to *reach* some target states, whereas the goal of player 2 is to make sure this does not happen.)