1)

A) Optimal substructure: The less students that participate in steps, the less switches. That combined with finding the largest sequences of steps a student can take

B) Greedy Algorithm: The greedy algorithm need to use is an optimal switching sequence. This can be done by finding the earliest largest sequence of steps. So starting at step 1, you find the largest sequence of steps that any one student can take. Then at the end of that sequence you you look for the next largest sequence any student can take, and you continue in that manner until all steps are taken.

D) The runtime complexity where n is the numSteps, and s is the numStudents, since there is the initial while loop up to n, and inside of that while loop there is a second while loop that will iterate at most s times, meaning it is in sn^2 $O(sn^2)$

E)

Algorithm: G_1, G_2, G_3 . . ., G_n

Optimal: O_1, O_2 O_3 . . ., O_n

Assuming that the greedy algorithm has i students, and the optimal solution has j students, where $i > j$.

There is a value k where the algorithm and the optimal solution don't have the same number of switches.

2)

A) Using a modified Dijkstra's algorithm that instead of finding the shortest path to all nodes given a source node, you only find the shortest path to the destination node. Given that information, you just need to implement the frequency at which trains run, and combining the two, you will
B) The time complexity of the algorithm would be V^2, as since the algorithm relies on a slightly modified dijkstra's which is also V^2
C) The algorithm in the method shortestTime is Dijkstra's shortest path algorithm.
D) Since the current methods in the file account for travel time taken between each station, but not for the actual time required to wait between trains arrivals, we need to add the frequency of trains to the equation.
E) The time complexity of shortestTime is $O(E + V^2)$, or $O(V^2)$ as V^2 dominates the runtime but if you were to use something like an adjacency list and a binary heap you could cut it down to $O(E \log V)$