

1A) The goal of this problem is to schedule students to complete all the steps with the minimum amount of switching possible. This can be achieved by using students who are able to do the most consecutive steps.

1B) Start by finding a student that can do the first step and then the max consecutive steps that the student can do. Then keep repeating the process over until all the steps have been scheduled to a student. If there are multiple students that can do the same max consecutive steps take the first student that was found.

1D) Runtime for the algorithm is $O(m \cdot n)$. The worst-case scenario would be if all the students could do every step, then we would have to check every step(n) for each student(m).

1E) Let us say there is ALG which represents my algorithm and OPT which represents an optimal algorithm. Now let's assume that ALG gives a solution of a_1, a_2, \dots, a_k and x switches and OPT gives us a solution of o_1, o_2, \dots, o_k and y switches and $y < x$. Let's say that until the i th index both algorithms are the same. If we use cut and paste we get $a_1, a_2, \dots, a_i, o_{i+1}, o_{i+2}, \dots, o_k$. Both algorithms are designed to take the students that do the most consecutive steps so replacing a_i with o_i will result in the same number of switches which means ALG is just as good as OPT. This contradicts $y < x$ which means that ALG has the most optimal solution.

2A) The algorithm used is a modified version of Dijkstra's algorithm that was altered to take into account of the wait time. The Dijkstra's algorithm finds the shortest path between two nodes. The only modification needed is to find the wait time for the trains and add that to the total time required to get from the starting point and the end point. This would include the time it takes the first train to arrive at station and the time it takes for the next train to arrive. I got the idea for the Dijkstra's algorithm from <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

2B) In a worst case scenario the time complexity would be $O(V^2)$

2C) The shortest Time method uses Dijkstra's Algorithm

2D) The shortest Time algorithm would have to be altered to take into account the additional wait time when the train is not present. This was discussed in 2A.

2E) The Complexity of shortest Time is $O(V^2)$. The complexity can be reduced to $O((E) + V \log V)$ by using a Fibonacci heap. This information was obtained from https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm