Experiments Scheduling:

a) A given problems has Optimal Substructure Property if optimal solution of the given problem can be obtained by using optimal solutions of its subproblems.

The optimal solution is the solution which takes all the necessary steps in ascending order with the lowest number of switches. It would minimize the number of switches when the students would be scheduled for doing in the sequential steps. Example, we have m number student who has assigned p number steps. Now every student tries their steps but most effective and less time consuming are less switches. It can be optimized with optimal substructures.

b) The greedy algorithms would be most effective way to get the optimal solution. we use greedy choice among the students who can use lowest steps to complete the tasks in an ascending order. It would take a smaller number of switching if students do most of the steps in row base order.

c) Code is in the file PhysicsExperiment.java

d) The run time complexity is 0(n*m).

(e) **Greedy Algorithm**: S1, S2, S3,……………. S(n) with p number of the students.

Optimal solution: O (1), O (2), O (3),……..0(n) with k numbers of the students.

Where p>k. let a is that S(a)! =O(a). The algorithm will continue S (t) where it will switch if it does not switch student. The means it will give the results in order like: S1, S2, S3……. St, O (t+1), O (t+2),.., O (n). This will give the same number of switches.

Public, Public Transit

(a) I will use Dijkstra's algorithm to solve this problem that we have leant in class. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. I would use the algorithm to keep track the shortest path, all the stoppage, the destination that has shortest path to arrive. After gathering all the necessary information, I would use first matrix, frequency matrix, and the beginning time to find the next train on the shortest path that we found from Dijkstra's algorithm.

(b) Maximum complexity is O (V * V). V is number of vertices.

(c) The algorithm that this is implementing is Dijkstra's algorithm.

(d) By using the algorithm, I will keep track the shortest path. Also, using the array implementation, I would keep track my starting point to my destination with the shortest path.

(e) The current complexity of "shortest Time" is O (V * V). The runtime complexity of the optimal implementation is O (E log V), where E is the number of edges and V is the number of vertices. We could make “shortest Time” by implementing breadth first search and using Queue data structure.

f) Code in FastestRoutePublicTransit.java

g) Code in FastestRoutePublicTransit.java