Taeyong Lee

1. Experiment Scheduling

   a) The Optimal solution is for scheduling students to complete all steps with smallest number of switch. We schedule student who can do largest number of consecutive steps and then schedule next student who can do 2nd largest number of consecutive steps but doesn't do same steps already first student did before. Repeat this until all steps are done.

   b) Greedy algorithm is algorithm that make the optimal selection for every moment. This algorithm make us schedule student who can do largest consecutive steps starting from current step for every step. So, solution of this problem will have least number of switching student.

   c) Please see PhysicsExperiment.java

   d) O(nm) for n student and m steps

   e) Assume that there's algorithm ALG and optimal solution OPT

      ALG<S1, S2,.....Sa> with n switch

      OPT<S'1, S'2......Sa> with m switch

      n>m

      There's some number i that 1<i<a and is the smallest index index that satisfies Si != S'i. And Si and S'i can be different by making switch or only OPT making switch.

      If both are making switch, S1, S2,.....Si and S'1, S'2......Si are same and S1, S2,.....Si can be changed to S'1, S'2......Si. Result of this will be S1, S2,.....Si, S'i+1, S'i+2......Sa.

      If only OPT is making switch, Result will be S1, S2,.....Sb, S'b+1, S'b+2......Sa that b is some number less than i because ALG has less switch than OPT. Therefore, Greedy Algorithm gives optimal solution for this problem.

2. Public, Public Transit

   a) I used Dijkstra's algorithm with some modification to find shortest path between starting station and target station because it is good for searching shortest path between 2 vertices in the graph.

   b) O(V^2) – complexity of Dijkstra's algorithm and can be faster with complexity of O(E log V)

c) Dijkstra's algorithm

d) Due to original Dijkstra's algorithm can handle only one data per edge, I had to add another array to save previous data. So, I can keep backtracking my way from destination to starting station and I can find shortest path from starting station to destination.

e) Current complexity of "shortestTime" is O(V^2). If we use Fibonacci heap instead of adjacency matrix, "shortestTime" will be faster with complexity of O(E log V).