Holiday Special Write-up
Daniel Kopeloff

A) Describe the optimal substructure of this problem.

The optimal substructure of this problem is to find the cook that can perform the current step and the one next one without switching. By doing this it will work for any N cooks

B) Describe the greedy algorithm in plain words that could find an optimal way to schedule the volunteers for one recipe.

The greedy algorithm of this problem is to find the cook that can perform the current step and the next step. If you are able to do that then you are able to determine if a switch will be needed or not. This is greedy because we are able to determine what is best locally at the point of each individual step. If two cooks can both do the same job, then you would check their next steps. If they end up being able to complete the same exact steps, then we will choose the first one.

D) What is the runtime complexity of your greedy algorithm? Again, you don't need to factor in the setup of the signup table, just your scheduling algorithm.

The time complexity of the greedy algorithm is O(numCooks * numSteps) so O(n^2). This is because I have to go through each step with each cook to find the optimal schedule.

E) In your write-up file, based on your answer to part b, give a full proof that your greedy algorithm returns an optimal solution.

Proof by contradiction

Let A be the set containing the solution of my algorithm. Then Let B be the set containing an optimal solution. We can assume that B is a valid solution and contains valid elements in its solution. Since A != B we can know that they are different elements in their sets. In this example the sets would contain the order of the cooks for the recipe. B will have a different order then A. the best solution it can be is that it has no switches but if you take out one cook and change him then you have at least one switch. By having at least one switch it's just as good as the A algorithm. Since there cannot be a more optimal solution then B is not a more optimal solution and is either just as good or worse than A algorithm.