Due on February 15th, 2017
30 points total

General Directions: If you are asked to provide an algorithm, you should clearly define each step of the procedure, establish its correctness, and then analyze its overall running time (for this assignment, this means arguing why your algorithm achieves the target running time specified by the question). There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice.

All the answers must be typed, preferably using LaTeX. If you are unfamiliar with LaTeX, you are strongly encouraged to learn it. However, answers typed in other text processing software and properly converted to a pdf file will also be accepted. Before submitting the pdf file, please make sure that it can be opened using any standard pdf reader (such as Acrobat Reader) and your entire answer is readable. **Handwritten answers or pdf files that cannot be opened will not be graded and will not receive any credit.**

Finally, please read the detailed collaboration policy given on the course website. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

Problem 1 (15 points)
After spending several exhausting weeks attempting to find locally highest points in Durham, our good friend Bob Bitfiddler is now planning a well-deserved vacation. While browsing online, he was fortunate enough to stumble upon a travel site offering an attractive international vacation package. For discounted flight fares, the deal works as follows: the travel site gives Bob a fixed sequence of n countries $S = C_1, C_2, \ldots, C_n$. First, Bob picks any country in the sequence where he wants to begin his trip. He must then proceed by flying from country to country according to the sequence, but can decide to travel back home whenever he likes. More formally, Bob determines his trip by picking some contiguous subsequence $S' = \langle C_s, C_{s+1}, \ldots, C_{e-1}, C_e \rangle$ of S, where $1 \leq s \leq e \leq n$.

Bob wants to be clever in selecting the countries that he will visit. He has an extra \$100 that he has decided not to spend during the vacation. Instead, by using exchange rates, he hopes to turn a profit on this money. We will model exchange rates in the following way:

1. Before Bob leaves for the trip, he will convert this \$100 to the native currency of the first country C_s that he is going to visit. This conversion is without change in value (i.e., he will have \$100 worth of currency in the first country he visits).

2. We define $\langle c_{1,2}, c_{2,3}, \ldots, c_{n-1,n} \rangle$ to be the $n-1$ *exchange factors* between the n countries in S, where each $c_{i,j} \in \mathbb{Q}^+$ gives the factor by which the value of Bob's assets change when he exchanges currency from country i to country j. (Note that $\mathbb{Q}^+$ denotes the set of positive rational numbers).

3. Once he returns home, he can convert the currency he has in the last country C_e that he visited back to USD without change in value.

To make this clear, consider the following examples. If Bob decides to just visit a single country and then fly back, i.e. if the subsequence he picks is $S' = \langle C_s \rangle$ for some $1 \leq s \leq n$, then the value of his \$100 is not change, i.e. he makes no profit. On the other hand, suppose that Bob decides to visit the first three countries, i.e. $S' = \langle C_1, C_2, C_3 \rangle$, and then fly home. In this case, the value of his initial \$100 changes to $\$100 c_{1,2} c_{2,3}$. Depending on the values of $c_{1,2}$ and $c_{2,3}$, he either made a profit or lost money.

Bob would like to pick his trip sequence S' such that he maximizes his profit from the \$100 that he started with. Give an algorithm that determines such a sequence in $O(n)$ time.

Problem 2 (15 points)
It is Game Day and all roads lead to Cameron! The university has identified a set of roads that get congested leading up to the game and wants to set up traffic monitors at intersections to ensure the smooth flow of traffic on these roads. A monitor placed at an intersection can monitor traffic flow on all the roads meeting at the intersection. Your prowess in COMPSCI 330 lands you the job of deciding the minimal number of monitors needed to ensure that all roads are monitored. (You can assume that every road connects two intersections with no intersection in between.) This

problem seems difficult at first glance, but then you observe that the network of roads that need to be monitored is acyclic! Can you now come up with a linear-time algorithm to solve the problem? (Don't worry about reporting where the monitors need to be placed.)