EECS 3101 Fall '17: Design and Analysis of Algorithms

Assignment 4

Weight: 5%, Due: Dec 5, in the drop box by 5 pm

Notes:

1. The assignment can be handwritten or typed. It MUST be legible.
2. You must do this assignment individually.
3. Submit this assignment only if you have read and understood the policy on academic honesty on the course web page. If you have questions or concerns, please contact the instructor.
4. Use the dropbox near the EECS main office to submit your assignments. No late submissions will be accepted. Please do not send files by email unless you have the instructor's permission to do so.

Problem 1

Consider the problem of cutting rods in assignment 2. Using the same cost model, design a greedy algorithm for cutting out rods of given lengths from a given rod stock. You are not given cut locations in this version, but rather the pieces that you need to produce.

Prove that your algorithm is optimal and analyze the running time of your algorithm.

Problem 2

In the old days there was barter for goods. One could do this to avoid monetary transactions and at a state level barter circumvented the need for currency conversion and appropriate conversion rates. Imagine a scenario where there are n states $s_1, ..., s_n$ and a state needed to exchange one item for another (e.g. state i produced wheat but wanted fabric, state j produced fabric but wanted iron etc). There are k commodities in total that were being produced and required over all states.

Given queries of the form "Can state i get fabric from state j?", your job is to produce an algorithm for answering such queries efficiently.

There are some rules to follow. Each state can produce up to c commodities and require up to c commodities, where c is a constant. Also, The state designated buyer from state i cannot cross through another state j with a commodity that state j is not interested in procuring. So if j wanted to barter iron for potatoes, the buyer from state i could enter j with potatoes and leave with the iron. However, it would only leave j and enter a state $k \neq i$ if k required iron, and so on.

Obviously there could be many ways of going from state i to state j and only some of them are valid according to the above rules.

First, imagine the case where all valid ways in which state i could get fabric from state j result in the same "exchange rate" between the bartered items.

Second, explain how you you would find the best exchange rate if different paths had different exchange rates. Analyze the running times of your algorithms.