CS3230: Design and Analysis of Algorithms (Fall 2014)

Tutorial Set #11

[For discussion during Week 13]

S-Problems are due (outside Prof. Leong's office): Friday, 7-Nov, before noon.

OUT: 5-Nov-2014 **Tutorials:** Tue & Wed, 11, 12 Nov 2014

IMPORTANT: Read "Remarks about Homework".

Submit solutions to S-Problem(s) by deadline given above.
Prepare your answers to all the D-Problems in every tutorial set.

When preparing to present your answers,

- Think of a CLEAR EXPLANATION
- Illustrate with a good worked example;
- Describe the main ideas,
- Can you sketch why the solution works;
- Give analysis of running time, if appropriate
- Can you think of other (perhaps simpler) solutions?

Helpful Hints Series:
Please note that getting the right recursion is quite important for dynamic programming algorithms. After that the pseudocode is more or less clear.

Routine Practice Problems *-- do not turn these in -- but make sure you know how to do them.*

R1. Does a greedy approach work for the Knapsack problem? Yes/No?

R2. Does greedy approach always work for the coin change problem? Yes/No?

R2. The total number of sub-problems involved in a dynamic programming solution can be exponential? Yes/No?

R3. Dynamic programming algorithms are helpful for Unix Diff? Yes/No?

S-Problems: (To do and submit by due date given in page 1)
Solve this S-problem(s) and submit for grading.

IMPORTANT: Write your NAME, Matric No, Tutorial Group in your Answer Sheet.

S1. [Maximum difference]

We are given a sequence of n > 1, non-negative distinct real numbers $p_1, p_2, \ldots, p_n$ (not necessarily in the increasing order). We are supposed to determine

$$\max\{p_j - p_i \mid 1 \le i < j \le n\}.$$

Write a dynamic-programming algorithm (idea and pseudocode) for this task.

D-Problems: Solve these D-problems and prepare to discuss them in tutorial class. You may be called upon to present your solution *or your best attempt at a solution*. Your solution presentation does NOT need to be fully correct, given your best attempt. The TA will help clarify and correct any issues or errors.

D1. [Coin change revisited]

Write an algorithm which takes as input an array d of n denominations with d[n] > d[n-1] > … > d[1] = 1 and an amount A. It outputs the minimum number of coins used for amount A, using denominations in d. Write the idea and the pseudocode of the algorithm.

D2. [Catalan numbers and matrix multiplication]

The n-th Catalan number is defined to be:

$$C_n = \sum_{i=1}^{n} C_{i-1} C_{n-i}$$

for all numbers n ≥1, where we define $C_0 = 1$. Prove that the number of ways to group the product $(M_1\ M_2\ \ldots\ M_n)$, n ≥1 (here each M_i is a matrix) is equal to C_{n-1} . For example for product of three matrices $(M_1\ M_2\ M_3)$, different ways to group are $(M_1\ (M_2\ M_3\ ))$ and $((M_1 M_2)\ M_3\ )$.

D3. [Robot walk]

A robot can take steps of 1 meter, 2 meter and 4 meter. Write a dynamic programming algorithm (idea and pseudocode) to calculate the number of ways the robot can walk n meters. Note that you need to take the order into account, that is say when n=7, walk 1,2,4 is different from walk 4,1,2.

D4. [Optimizing choice]

You are given natural numbers $l_1, l_2, \dots, l_n$, h_1 $h_2, \dots, h_n$, such that for $i \in \{1, \dots n\}$: $1 \leq l_i < h_i$. Write a dynamic programming algorithm to find maximum value of $\sum_{i=1}^{n} p_i$, under the following rules:

1. For each $i \in \{1, \dots n\}$: $p_i = h_i$ or $p_i = l_i$ or $p_i = 0$.
2. For each $i \in \{1, \dots n-1\}$: If $p_{i+1} = h_{i+1}$ then $p_i = 0$.