

COMP4500/7500 Advanced Algorithms & Data Structures

Tutorial Exercises 7 (2014/2)*

School of Information Technology and Electrical Engineering, University of Queensland

September 8, 2014

This material aims to familiarise you with some aspects of shortest paths algorithms (CLRS Chapter 25 [2nd, 3rd]; CLR Chapter 26 [1st]) and greedy algorithms. A good treatment of greedy algorithms may be found in CLRS Chapter 16; CLR Chapter 17.

1. (CLRS Exercise 25.2-4, p699 [3rd] ; CLR Exercise 26.2-2)

The Floyd-Warshall algorithm (CLRS p695 [3rd]) requires $\Theta(n^3)$ space because we compute $D^{(k)}$ for $k = 0, 1, \dots, n$ and each matrix has n^2 elements.

FLOYD-WARSHALL(W)

```
1 // W is the weight matrix
2 n = W.rows
3  $D^{(0)} = W$ 
4 for k = 1 to n
5     let  $D^{(k)} = (d_{ij}^{(k)})$  be a new  $n \times n$  matrix
6     for i = 1 to n
7         for j = 1 to n
8              $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
9 return  $D^{(n)}$ 
```

Show that the following procedure, which simply drops all the superscripts, is correct, and thus only $\Theta(n^2)$ space is required. You may assume that W has no negative-weight cycles.

FLOYD-WARSHALL'(W)

```
1 n = W.rows
2 D = W
3 for k = 1 to n
4     for i = 1 to n
5         for j = 1 to n
6              $d_{ij} = \min(d_{ij}, d_{ik} + d_{kj})$ 
7 return D
```

2. (CLRS Exercise 25.2-6, p700 [3rd]; CLR Exercise 26.2-5)

How can the output of the Floyd-Warshall algorithm be used to detect the presence of a negative-weight cycle?

3. (CLRS Exercise 16.1-4, p422 [3rd], 16.1-3 [2nd]; CLR Exercise 17.1-2 [1st])

Suppose that we have a set of activities to schedule among a large number of lecture halls, where any activity can take place in any lecture hall. Each activity has a given start and finish time. We wish to schedule all the activities using as few lecture halls as possible.

- Give an efficient greedy algorithm to determine which activity should use which lecture hall.
- Justify that your algorithm has the greedy-choice property.
- Give the worst-case time complexity of your algorithm.

(This is also known as the *interval-graph colouring problem*. We can create an interval graph whose vertices are the given activities and whose edges connect incompatible activities. The smallest number of colours required to colour every vertex so that no two adjacent vertices are given the same colour corresponds to finding the fewest lecture halls needed to schedule all the given activities.)

*Copyright © reserved September 8, 2014

4. (CLRS Exercise 16.1-3, p422 [3rd]; 16.1-4 [2nd]; CLR Exercise 17.1-3 [1st])

Not just any greedy approach to the activity selection problem produces a maximum-size set of mutually compatible activities.

- (a) Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work.
- (b) Do the same for the approach of always selecting the activity that overlaps the fewest other remaining activities.

5. (CLRS Exercise 16.2-1, p427 [3rd]; CLR Exercise 17.2-1)

The **0-1 knapsack problem** is as follows. A thief finds N items; the i th item is worth v_i dollars and weighs w_i kilograms, where v_i and w_i are integers. He wants to take as valuable load as possible, but can carry at most W kilograms in his knapsack for some integer W . What items should he take? (It is called **0-1** because each item is either taken or left behind; a fraction of an item cannot be taken. This is a **hard** problem to solve optimally.)

In the **fractional knapsack problem** the setup is the same, but the thief can take fractions of items, rather than having to make a binary (0-1) choice for each item. A sample item for the 0-1 problem might be a gold bar, while for the fractional problem it might be gold dust.

Prove that the fractional knapsack problem has the greedy-choice property. In the fractional knapsack problem we may take fractions of an item to add to the knapsack. The greedy strategy is to process the items in non-increasing order of their value per unit weight, adding them to the knapsack until it is full. For the last item we may only add a fraction of its available weight before filling the knapsack.

6. (CLRS Exercise 16.2-4, p427 [3rd]; CLR Exercise 17.2-4)

Professor Midas drives an automobile from Brisbane to Sydney along the New England Highway. Her car's petrol tank, when full, holds enough petrol to cover k kilometres, and her map gives the distances between petrol stations on the route. The professor wishes to make as few stops for petrol as possible along the way. You may assume her petrol tank is initially full.

- (a) Give an efficient method by which Professor Midas can determine at which petrol stations she should stop. (This is the easy part.)
- (b) Prove that your strategy yields an optimal solution.