FIT1008 Introduction to Computer Science (FIT2085 for Engineers)

Tutorial 10

Semester 1, 2019

Objectives of this tutorial

- To understand Dynamic Programming.

Exercise 1 *

Write down an algorithm to compute the n-th Fibonacci number using Dynamic Programming. How does this compare in terms of time complexity to the recursive and tail recursive implementations done in the class?

Exercise 2 *

You are presented with a row of 5 coins with values $7, $2, $10, $12, $5. Pick up the largest amount of money from the row of coins, with the constraint that you cannot pick up any two adjacent coins.

How can you design an algorithm to solve the problem for a row of n coins of arbitrary values?

Exercise 3 *

Using dynamic programming, solve the knapsack problem given by the table below. The knapsack capacity is 17 kg.

Item	**1**	**2**	**3**	**4**	**5**
Value ($)	4	5	10	11	13
Weight (kg)	3	4	7	8	9

Exercise 4 *

In solving the knapsack, assume that you are now allowed to take duplicates, i.e. each item can be selected multiple times, as long as the knapsack capacity is not exceeded. Discuss how you would modify the Dynamic Programming algorithm given in the lecture to solve the problem when duplicate are allowed.

Exercise 5

- Write a **recursive** algorithm to compute the binomial coefficients, given by the following formula:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Assume $n \geq k$, with $n > 0$, $k > 0$, $n, k \in \mathbb{Z}$. Note that

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

- Write a **Dynamic Programming** algorithm to compute the binomial coefficients.

Compare the complexity of the two implementations.