**SC2001/CE2101/CZ2101 ALGORITHM DESIGN AND ANALYSIS**

Project 3: Dynamic Programming

We have a knapsack of capacity weight *C* (a positive integer) and *n* types of objects. Each object of the ith type has weight *wi* and profit *pi* (all *wi* and all *pi* are positive integers, i = 0, 1, …, *n-1*). There are unlimited supplies of each type of objects. Find the largest total profit of any set of the objects that fits in the knapsack.

Let P(*C*) be the maximum profit that can be made by packing objects into the knapsack of capacity *C*.

1. Give a recursive definition of the function P(*C*).
2. Draw the subproblem graph for P(14) where *n* is 3 with the weights and profits given below.

0 1 2

|  |  |  |
| --- | --- | --- |
| 4 | 6 | 8 |
| 7 | 6 | 9 |

**wi**

**pi**

1. Give a dynamic programming algorithm to compute the maximum profit, given a knapsack of capacity *C*, *n* types of objects with weights *wi* and profits *pi* using the bottom up approach.
2. Code your algorithm in a programming language
   1. show the running result of P(14) with weights and profits given in (2).
   2. Show the running result of P(14) with weights and profits given below.

0 1 2

|  |  |  |
| --- | --- | --- |
| 5 | 6 | 8 |
| 7 | 6 | 9 |

**wi**

**pi**

4a.

A screenshot of a computer

Description automatically generated

4b.

A screenshot of a computer

Description automatically generated