Department of Mathematical Sciences
**CARNEGIE MELLON UNIVERSITY**

# OPERATIONS RESEARCH II 21-393

Homework 1: Due Monday September 21.

**Q1** Solve the following knapsack problem:

$$\begin{array}{lcccccccc} \text{maximise} & 4x_1 & + & 8x_2 & + & 13x_3 & & \\ \text{subject to} & & & & & & & \\ & 3x_1 & + & 4x_2 & + & 5x_3 & \leq & 16 \end{array}$$

$$x_1, x_2, x_3 \geq 0 \text{ and integer.}$$

**Q2** An $m \times n$ rectangle of wood is to be cut into smaller rectangles. An $a \times b$ rectangle is worth $m_{a,b}$. The machine that cuts rectangles can only cut full length or full width. I.e. if after cutting there is an $x \times y$ rectangle then the machine can cut it into two rectangles $z \times y$ and $(x - z) \times y$ for some $z$ or into two rectangles $x \times z$ and $x \times y - z$.
Describe a dynamic programming algorithm for finding the way of cutting into pieces that maximises the total value of the rectangles produced.
**Q3** Consider a 2-D map with a horizontal river passing through its center. There are $n$ cities on the southern bank with $x$-coordinates $a(1)...a(n)$ and $n$ cities on the northern bank with $x$-coordinates $b(1)...b(n)$. You want to connect as many north-south pairs of cities as possible with bridges such that no two bridges cross. When connecting cities, you can only connect city $i$ on the northern bank to city $i$ on the southern bank. Construct a Dynamic Programming solution to this problem. (You can assume that $a(1) < a(2) < \cdots < a(n)$, but you **cannot** assume that $b(1) < b(2) < \cdots < b(n)$. If both sequences are increasing, then the problem is trivial).