Department of Mathematical Sciences
Carnegie Mellon University

21-393 Operations Research II
Test 1

Name:_____

Problem	Points	Score
1	35	
2	35	
3	30	
Total	100	

Q1: (35pts)
(a) Fill in the last column of the table below for solving the following knapsack problem and produce an optimal solution:

$$\begin{array}{lrcrcrcl} \text{maximise} & 3x_1 & + & 7x_2 & + & 20x_3 & & \\ \text{subject to} & 2x_1 & + & 3x_2 & + & 5x_3 & \leq & 12 \end{array}$$

$x_1, x_2, x_3 \geq 0$ and integer.

w	$f_1(x_1)$	b_1	$f_2(x_2)$	b_2	$f_3(x_3)$	b_3
0	0	0	0	0		
1	0	0	0	0		
2	3	1	3	0		
3	3	1	7	1		
4	6	1	7	1		
5	6	1	10	1		
6	9	1	14	1		
7	9	1	14	1		
8	12	1	17	1		
9	12	1	21	1		
10	15	1	21	1		
11	15	1	24	1		
12	18	1	28	1		

(b) Read off the solution to the problem below from the above table.

$$\begin{array}{lrcrcrcl} \text{minimise} & 2x_1 & + & 3x_2 & + & 5x_3 & & \\ \text{subject to} & 3x_1 & + & 7x_2 & + & 20x_3 & \geq & 20 \end{array}$$

$x_1, x_2, x_3 \geq 0$ and integer.

(c)

$B.V.$	x_1	x_2	x_3	x_4	ξ_1	ξ_2	RHS
x_0	1		1				7
x_2	1/2	1	$-1/4$				7/4
x_4	17/2		$-3/4$	1			35/4

Suppose that the above tableau represents the optimal basic feasible solution to a pure interger programming problem. Derive a Gomory cut and indicate where the first dual simplex pivot should be.

Q2: (35pts)
You are given the task of cutting up an $n \times n$ square into rectangles. The rectangle with corners $(a, b), (c, d)$ has value $v(a, b, c, d)$, where $a < c$ and $b < d$. A cut must partition the current rectangle into two rectangles. One will be sold at value and the other (if non-empty) will be used for continued cutting. All cuts must be horizontal or vertical and go through integer points. Give a Dynamic programming formulation for the problem of maximising the value of the set of rectangles produced.

Q3: (30pts)
Formulate the following as an integer program: n jobs need to be done and there are m people available. s teams $T_1, T_2, \ldots, T_s \subseteq [m]$ have been identified. Let $a_{i,j} = 1$ if team T_i is capable of completing job j. We want to determine a collection of disjoint teams that can cover every job. We want to do this in a way that involves as few teams as possible.