Department of Mathematical Sciences
CARNEGIE MELLON UNIVERSITY

OPERATIONS RESEARCH II 21-393

Homework 1: Due Monday September 10.

Q1 Solve the following knapsack problem:

$$\begin{array}{lccccccc} \text{maximise} & 4x_1 & + & 8x_2 & + & 15x_3 & & \\ \text{subject to} & & & & & & & \\ & 3x_1 & + & 4x_2 & + & 5x_3 & \leq & 19 \end{array}$$

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

Solution

w	f_1	δ_1	f_2	δ_2	f_3	δ_3
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	4	1	4	0	5	0
4	4	1	8	1	8	0
5	4	1	8	1	15	1
6	8	1	10	0	15	1
7	8	1	13	1	15	1
8	8	1	16	1	20	1
9	12	1	16	1	23	1
10	12	1	18	1	30	1
11	12	1	21	1	30	1
12	16	1	24	1	30	1
13	16	1	24	1	35	1
14	16	1	26	1	38	1
15	20	1	29	1	45	1
16	20	1	32	1	45	1
17	20	1	32	1	45	1
18	24	1	34	1	50	1
19	24	1	37	1	53	1

Solution: $x_1 = 0, x_2 = 1, x_3 = 3$. Maximum = 53.
Start with $x_1 = x_2 = x_3 = 0$. $\delta_3(19) = 1$ and so we add one to x_3. We have used up 5 units of the knapsack. There are 14 units left. $\delta_3(14) = 1$ and so we add one to x_3. We use up another 5 units and so we are left with 9. $\delta_3(9) = 1$. We add one more to x_3. There are now 4 units in the knapsack. $\delta_3(4) = 0$ and so we move to column 2. $\delta_2(4) = 1$ and so we add one to x_2. This reduces the knapsack capacity to 0, We have $\delta_2(0) = 0$ and we move to column 1. $\delta(1) = 0$ and we are done.

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.
Solution First assume that if you make a vertical cut then the piece to the right cannot be cut further. Similarly, if you make a horizontal cut then the higher piece cannot be cut further. Then if $f(M, N)$ denotes the maximum that can be obtained from the rectangle with corners $(0, 0)$ and (M, N),

$$f(M,N) = \max \begin{cases} \max\{f(M, N-y) + m_{M,y} : 0 < y \leq N\} & \text{horizontal cut} \\ \max\{f(M-x, N) + m_{x,N} : 0 < x \leq M\} & \text{vertical cut} \end{cases}$$

If you are allowed tocut up both pieces then

$$f(M,N) = \max \begin{cases} \max\{f(M, N-y) + f(M,y) : 0 < y \leq N\} & \text{horizontal cut} \\ \max\{f(M-x, N) + f(x,N) : 0 < x \leq M\} & \text{vertical cut} \\ m_{M,N} & \text{no cut} \end{cases}$$

Q3 We are given $2n$ sets $D_1, D_2, \ldots D_n$ and $R_1, R_2, \ldots, R_n$ where n is even. Also, $|D_i| + |R_i| = m$ for $i = 1, 2, \ldots, n$. Find an algorithm that will check to see if the following is possible: Find a set $I \subseteq [n]$, $|I| = n/2$ such that

$$\sum_{i \in I} |D_i| \geq \sum_{i \in I} |R_i| \text{ and } \sum_{i \notin I} |D_i| \geq \sum_{i \notin I} |R_i|.$$

Your algorithm should run in time polynomial in m, n.
Solution: For $I \subseteq [n]$ let $D_I = \sum_{i \in I} |D_i|$ and $R_I = \sum_{i \in I} |R_i|$. Then let

$$f_{k,\ell}(a,b,c,d) = \begin{cases} 1 & \exists I \subseteq [k] : |I| = \ell \text{ and } D_I = a, D_{[k] \setminus I} = b, R_I = c, R_{[k] \setminus I} = d \\ 0 & otherwise \end{cases}$$

Then we have the recurrence

$$f_{k+1,\ell}(a,b,c,d) =$$
$$\begin{cases} 1 & f_{k,\ell-1}(a - |D_{k+1}|, b, c - |R_{k+1}|, d) + f_{k,\ell}(a, b - |D_{k+1}|, c, d - |R_{k+1}|) \geq 1 \\ 0 & otherwise \end{cases}$$

Having computed $f_{n,n/2}$ we can check to see whether or not there exist $a \geq c, b \geq d$ such that $f_{n,n/2}(a,b,c,d) = 1$. This takes $O((mn)^4 n^2)$ time, since this is the number of function evaluations we need to compute. Note that $D_{[n]} + R_{[n]} = mn$. (We can save a bit of time by only evaluating $f_{k,\ell}$ when $a + b + c + d = km$.)