Answers to CS / MCS 401 Week 12 Exercises (Fall, 2007)

(except exercises 15-1 and 15-6, to be discussed in class)

Exercise 15.4–1

The matrices (c_{ij}) and (b_{ij}) are constructed using the algorithm in the textbook (page 353). We obtain

		0	1	2	3	4	5	6	7	8	9
	x		**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**
	y										
0	0	0	0	0	0	0	0	0	0	0	0
1	**1**	0	0 ↑	1 ↖	1 ←	1 ↖	1 ↖	1 ←	1 ↖	1 ↖	1 ←
2	**0**	0	1 ↖	1 ↑	2 ↖	2 ←	2 ←	2 ↖	2 ←	2 ←	2 ↖
3	**0**	0	1 ↖	1 ↑	2 ↖	2 ↑	2 ↑	3 ↖	3 ←	3 ←	3 ↖
4	**1**	0	1 ↑	2 ↖	2 ↑	3 ↖	3 ↖	3 ↑	4 ↖	4 ↖	4 L
5	**0**	0	1 ↖	2 ↑	3 ↖	3 ↑	3 ↑	4 ↖	4 ↑	4 ↑	5 ↖
6	**1**	0	1 ↑	2 ↖	3 ↑	4 ↖	4 ↖	4 ↑	5 ↖	5 ↖	5 ↑
7	**0**	0	1 ↖	2 ↑	3 ↖	4 ↑	4 ↑	5 ↖	5 ↑	5 ↑	6 ↖
8	**1**	0	1 ↑	2 ↖	3 ↑	4 ↖	5 ↖	5 ↑	6 ↖	6 ↖	6 ↑

A longest common subsequence of x and y has length c[8,9] = 6. We can read of the sequence by starting in the lower right corner and following the arrows, till we reach row 0 or column 0. Each time we move from square i,j to the upper left, we must have $x_i = y_j$, and we prepend x_i to the subsequence of $\boldsymbol{x}$ and y_j to the subsequence of $\boldsymbol{y}$ that we are building up. We obtain subsequences $x_1x_2x_3x_4x_6x_7$ of $\boldsymbol{x}$ and $y_2y_3y_6y_7y_8y_9$ of $\boldsymbol{y}$ which are provide a longest common subsequence of $\boldsymbol{x}$ and $\boldsymbol{y}$ — specifically, the subsequence **1,0,0,1,1,0**.

Problem 15-7 Reordering the jobs if necessary, we may assume $d_1 \le d_2 \le . . . \le d_n$. Then we may assume that, once the jobs are selected, they are run in order of increasing deadline. (If this schedule is not feasible, no other one can possibly be.)

Let $r[i,j]$ = maximum profit that can be earned if we are restricted to jobs chosen from $\{a_1, ..., a_i\}$ and if all jobs must finish by time j. We are interested in $r[n, d_n]$.

Note $r[i, 0] = 0$ for all i,
$r[0, j] = 0$ for all j.

Now consider $r[i,j]$.

i) If we don't select job a_i, then $r[i, j] = r[i-1, j]$. (Job a_i didn't help.) Not choosing job a_i is always feasible.

ii) Say we do choose job a_i. Then any jobs chosen from $\{a_1, ..., a_{i-1}\}$ must finish by time $\min(j, d_i) - t_i$, in order job a_i may finish by time $\min(j, d_i)$. Thus

$$r[i, j] = p_i + r[i-1, \min(j, d_i) - t_i]$$

Note that choosing job a_i is feasible only if $t_i \le \min(j, d_i)$.

Combining (i) and (ii), we obtain

$$r[i, j] = \begin{cases} r[i-1, j] & \text{if } t_i > \min(j, d_i), \\ \max\big(r[i-1, j],\ p_i + r[i-1, \min(j, d_i) - t_i]\big) & \text{otherwise} \end{cases}$$

Now we can compute all the $r[i, j]$ in $\Theta(nd_n)$ time by

```
Initialize: r[i, 0] = 0 for i = 1,...,n, and r[0, j] = 0 for j = 0,...,d_n.
for ( i = 1, 2, ..., n )
    for ( j = 1, 2,..., d_n )
        Compute r[i, j] by the formula above.
```

The maximum profit is $r[n, d_n]$. To find the schedule that produces the maximum profit, note a_n is chosen if and only if $r[n,d_n] > r[n-1,d_n]$. We could print the list of jobs chosen by the recursive function *print_jobs*( ), which is invoked initially as *print_jobs*(n, d_n).

```
print_jobs( i, j)
    if ( i == 0 )
        return;
    if ( r[i,j] > r[i–1, j] )
        print a_i;
        print_jobs(i–1, min(j, d_i) –t_i);
    else
        print_jobs(i–1, j);
```

Exercise Q.

$d^{0}_{1,10} = \infty$

$d^{1}_{1,10} = \infty$

$d^{2}_{1,10} = \infty$

$d^{3}_{1,10} = 60$ short$_3$(1,10) = 1,3,10

$d^{4}_{1,10} = 58$ short$_4$(1,10) = 1,2,4,10

$d^{5}_{1,10} = 55$ short$_5$(1,10) = 1,2,5,3,10

$d^{6}_{1,10} = 55$ short$_6$(1,10) = 1,2,5,3,10

$d^{7}_{1,10} = 50$ short$_7$(1,10) = 1,6,7,5,3,10

$d^{8}_{1,10} = 48$ short$_8$(1,10) = 1,6,7,5,4,8,10

$d^{9}_{1,10} = 46$ short$_9$(1,10) = 1,2,9,8,10

$d^{10}_{1,10} = 46$ short$_{10}$(1,10) = 1,2,9,8,10