Answers to CS/MCS 401 Exercise Set #5

(Exercises 15.2-1, 15-7, K, and L)

15.2-1 We initialize all $m_{ii} = 0$ and then compute the remaining m_{ij} and v_{ij} in order of increasing $j–i$, using the formula given in the text and in class (The text uses s_{ij} in place of v_{ij}), obtaining

i	j	m_{ij}	v_{ij}	i	j	m_{ij}	v_{ij}
1	2	150	1	1	4	405	2
2	3	360	2	2	5	2430	2
3	4	180	3	3	6	1770	4
4	5	3000	4	1	5	1655	4
5	6	1500	5	2	6	1950	2
1	3	330	2	1	6	2010	2
2	4	330	2				
3	5	930	4				
4	6	1860	4				

Since $v_{16} = 2$, in the optimal order the last multiplication comes after M_2. In other words, the optimal order has the form $(M_1 \times M_2) \times (M_3 \times M_4 \times M_5 \times M_6)$. Since v_{36}, the optimal order must have the form $(M_1 \times M_2) \times ((M_3 \times M_4) \times (M_5 \times M_6))$. In fact, this represents all we can say about the optimal order, since the relatively order of evaluating the independent operands $M_1 \times M_2$, $M_3 \times M_4$, and $M_5 \times M_6$ does not effect the number of scalar multiplication performed.

Problem 15-7 Reordering the jobs if necessary, we may assume $d_1 \le d_2 \le \ldots \le d_n$. 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

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

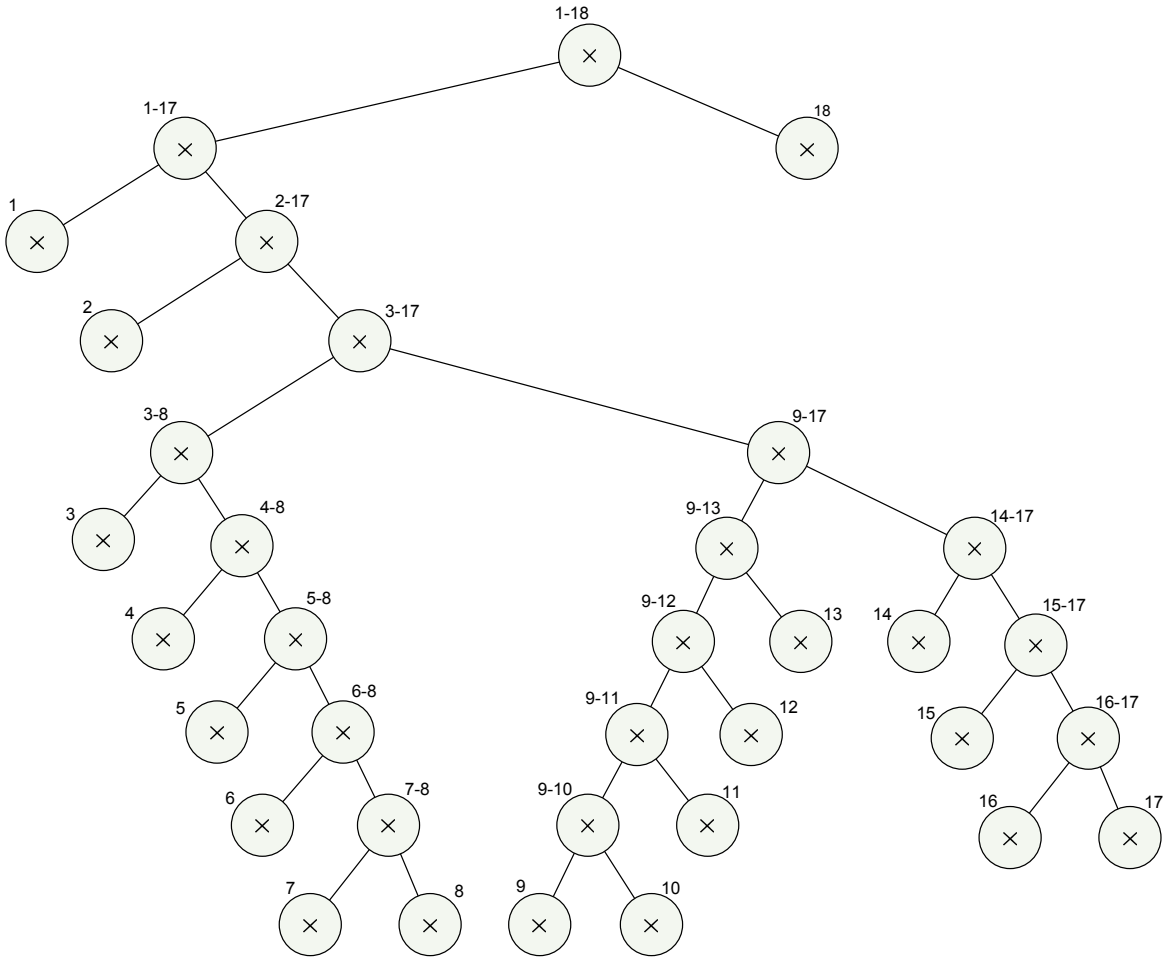
Now we can compute all the $r[i,j]$ in $\Theta(nd_n)$ time by

```
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);
```

K. The parse tree representing the optimal order of multiplication is shown below. The label i–j above a node indicates that the node corresponds to the product $M_i \times M_{i+1} \times ... \times M_j$. (If $i = j$, the label is written simply as i.) The left and right children of the node labeled i–j are the nodes labeled i–v_{ij} and v_{ij}+1–j, respectively.



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

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

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

$d^3_{1,10} = 60$ $\quad short_3(1,10) = 1,3,10$

$d^4_{1,10} = 58$ $\quad short_4(1,10) = 1,2,4,10$

$d^5_{1,10} = 55$ $\quad short_5(1,10) = 1,2,5,3,10$

$d^6_{1,10} = 55$ $\quad short_6(1,10) = 1,2,5,3,10$

$d^7_{1,10} = 50$ $\quad short_7(1,10) = 1,6,7,5,3,10$

$d^8_{1,10} = 48$ $\quad short_8(1,10) = 1,6,7,5,4,8,10$

$d^9_{1,10} = 46$ $\quad short_9(1,10) = 1,2,9,8,10$

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