SOLUTIONS FOR HOMEWORK 4

15.1-2

Consider the case $n = 3$ with the following price table

Length i	1	2	3
Price p_i	1	10	12
Density d_i	1	5	4

The greedy strategy will give $p_1 + p_2 = 11$, however our optimal solution is $p_3 = 12$.

15.1-3

Use the similar notations as our textbook, we have

$$r_n = \max_{1 \le i \le n} (p_i - c + r_{n-i})$$

and we have boundary condition $r_0 = 0$ (**this is important and you will lose points if you do not explicitly write it down**).

We also give the Pseudocode as Algorithm 1.

15.2-1

(Table omitted) The optimal parenthesization is $(A_1 A_2)((A_3 A_4)(A_5 A_6))$, with the minimum number of scalar multiplication 2010.

15.2-2

We borrow the implementation of Matrix-Multiply from our textbook, pages 371. Pseudocode is presented by Algorithm 2.

Algorithm 1: Rod-Cut(p, n, c)

```
$r_0 \leftarrow 0$
for $i = 1$ to $n$ do
    $r_i \leftarrow p_i$
    for $j = 1$ to $i - 1$ do
        $r_i \leftarrow \max\{r_i, p_j - c + r_{i-j}\}$
return $r_n$
```

Algorithm 2: Matrix-Chain-Multiply(A, s, i, j)

```
if  $i == j$ then
    return $A_i$
else
    $c \leftarrow s[i, j]$
    $M \leftarrow$ Matrix-Chain-Multiply$(A, s, i, c)$
    $N \leftarrow$ Matrix-Chain-Multiply$(A, s, c + 1, j)$
    return Maxtrix-Multiply$(M, N)$
```