HW#3
Due date: 5/14

1 Ex. 15.1-2 (5%)

2 Ex. 15.3-5 (5%)

3 Ex. 16.1-5
The analysis is closely related to that of Section 16.1 in the book.
You shall do steps 1, 3, and 4.
Step 1: Optimal substructure property (10%)
Write down an analysis that is analogous to that of p.416.
Let val$[i, j]$ denote the value of an optimal solution for the set S_{ij}.
Write down a recurrence for val$[i, j]$.
Step 2 – Recursive solution: memorization (This step may be omitted.)
Step 3 – Bottom-up tabulation (10%)
Step 4 – Constructing an optimal solution (5%)
Write down the pseudocode for steps 3 and 4 together and analyze its time and space complexities.

4 Ex 16.2-2 (10%)
The solution of this problem is posted publicly by the authors.
You are asked to read the solution, understand it, and rewrite the solution in your own words.

5 Ex. 16.2-6 (10%)
Hint: Do not sort the ratios $v_i/w_i, i = 1,2, \dots, n$. Find the median of the ratios and use it to partition the n items

6 Do Problem 16-1 a) (10%)
Do Problem 16-1 c) (5%)
Hint for part a)
Prove the following greedy-choice property.
Let c is the largest coin value such that $c \leq n$. Then, one coin of value c must be included in some optimal solution to making change for n cents.
Comment
When you prove this property by the technique of transformation, you will eventually find out that only the greedy choice yields the optimal solution.

7 [Programming exercise] 100%

Implement the dynamic programming solution for the problem of matrix-chain multiplication.

Basically, what you need to do is to write a C++ program that behaves like the pseudocode given in textbook pp.375,377, except that your program shall construct ***all*** the optimal ways to multiply the matrices.

Requirement

The exact implementation of the algorithm is up to you, as long as the following sample test is runnable.

Note that the sample test makes use of initializer_list constructors that are new in C++11, e.g. it uses the initializer_list **`{30,35,15,5,10,20,25}`** to construct a **`vector<int>`** object. Thus, you shall compile your program under C++11.

```
void testing_mcm(const vector<int>& p)
{
    static int test=0;
    cout << "Test " << ++test << "  ...\n";
    mcm(p);
    cout << endl;
}

int main()
{
    testing_mcm({30,35,15,5,10,20,25});
    testing_mcm({3,3,3,3,3});
    testing_mcm({2,2,3,2,2,3,3,2});
}
```

Sample output

```
Test 1  ...
15125 scalar multiplications
1 optimal way to multiply
((A1(A2A3))((A4A5)A6))
```

```
Test 2  ...
81 scalar multiplications
5 optimal ways to multiply
(A1(A2(A3A4)))
(A1((A2A3)A4))
((A1A2)(A3A4))
((A1(A2A3))A4)
(((A1A2)A3)A4)

Test 3  ...
66 scalar multiplications
10 optimal ways to multiply
(A1((A2A3)(A4(A5(A6A7)))))
(A1((A2A3)(A4((A5A6)A7))))
(A1(((A2A3)A4)(A5(A6A7))))
(A1(((A2A3)A4)((A5A6)A7)))
((A1(A2A3))(A4(A5(A6A7))))
((A1(A2A3))(A4((A5A6)A7)))
((A1((A2A3)A4))(A5(A6A7)))
((A1((A2A3)A4))((A5A6)A7))
(((A1(A2A3))A4)(A5(A6A7)))
(((A1(A2A3))A4)((A5A6)A7))
```