알고리즘

과제번호	13주차
날 짜	2018.12.06
학 번	201302395
이 름	류경빈

과제 1	Matrix Multiplication

```
MATRIX-CHAIN-ORDER (p)
n ← length[p] − 1
for i ← 1 to n
    do m[i, i] ← 0
for l ← 2 to n
    do for i ← 1 to n − l + 1
        do j ← i + l − 1
            m[i, j] ← ∞
            for k ← i to j − 1
                do q ← m[i, k] + m[k+1, j] + p_{i−1} p_k p_j
                    if q < m[i, j]
                        then m[i, j] ← q
                            s[i, j] ← k
return m and s
```

```java
public static void matrixChainOrder(ArrayList<Integer> p){
    m = new int[p.size()][p.size()];
    s = new int[p.size()][p.size()];
    int n = p.size()-1;
    for (int i = 1; i <= n; i++){
        m[i][i] = 0;
    }
    for (int l = 2; l <= n; l++){
        for (int i = 1; i <= n-l+1; i++){
            int j = i+l-1;
            m[i][j] = INFINITE;
            for (int k = i; k <= j-1; k++){
                int q = m[i][k] + m[k+1][j] + p.get(i-1) * p.get(k) * p.get(j);
                if (q < m[i][j]){
                    m[i][j] = q;
                    s[i][j] = k;
                }
            }
        }
    }
    printData(m, s);
}
```

MATRIX-CHAIN-ORDER 구현

```
PRINT-OPTIMAL-PARENS (s, i, j)
if i = j
    then Ai 를 출력
    else " ( " 를 출력
        PRINT-OPTIMAL-PARENS (s, i, s[i, j])
        PRINT-OPTIMAL-PARENS (s, s[i, j] + 1, j)
        " ) " 를 출력
```

```java
public static void printOptimalParens(int[][] s, int i , int j){
    if (i == j) {
        System.out.print("A" + (i));
    } else {
        System.out.print("(");
        printOptimalParens(s, i, s[i][j]);
        printOptimalParens(s, s[i][j] + 1, j);
        System.out.print(")");
    }
}
```

PRINT-OPTIMAL-PARENS 알고리즘 구현

결과 화면

```
MatrixMultiplication
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/ja
입력을 마칠 경우 0을 입력하세요.
1번 째 곱할 값 : 30
2번 째 곱할 값 : 35
3번 째 곱할 값 : 15
4번 째 곱할 값 : 5
5번 째 곱할 값 : 10
6번 째 곱할 값 : 20
7번 째 곱할 값 : 25
8번 째 곱할 값 : 0
---------------------------------
A(1)  = 30 x 35
A(2)  = 35 x 15
A(3)  = 15 x 5
A(4)  = 5 x 10
A(5)  = 10 x 20
A(6)  = 20 x 25
---------------------------------
     0 15750  7875  9375 11875 15125
    -1     0  2625  4375  7125 10500
    -1    -1     0   750  2500  5375
    -1    -1    -1     0  1000  3500
    -1    -1    -1    -1     0  5000
    -1    -1    -1    -1    -1     0
---------------------------------
    -1     1     1     3     3     3
    -1    -1     2     3     3     3
    -1    -1    -1     3     3     3
    -1    -1    -1    -1     4     5
    -1    -1    -1    -1    -1     5
    -1    -1    -1    -1    -1    -1
---------------------------------
Optimal solution : 15125
Optimal parens : ((A1(A2A3))((A4A5)A6))
```