EECS 336 Fall 2015
Homework Problem 5.2

Symbol multiplication table matching problem.

The idea is to divide p into all possible subsets, and examine each possible combination of subsets that constitute p to see if any resultant symbol matches with the given symbol.

Denote $(\alpha, \beta) = \gamma$ as the symbol multiplication between α and β according to T.

Step One: For $1 \leq i < j \leq m, X(i,j) \subset S$ is the set of symbols that are obtainable by the multiplication rule using symbols $\{p_i, ..., p_j\}$.

Step Two: Recurrence.

$X(i,j) = SET\{(\alpha,\beta)\}$, set of all possible symbol multiplications using $\alpha \in A, \beta \in B, \forall(\alpha,\beta) \in \{(X(i,i), X(i+1,j)), (X(i,i+1), X(i+2,j)), ..., (X(i,j-1), X(i,j))\}$.

Step Three: Base case $X(i,i) = p_i$

Step Four: Iterative DP

Algorithm 1 Iterative DP to find all symbols that are obtainable from p

```
procedure FindSymbol(p, t, T)
    for i = 1 to m do
        X(i, i) = p_i
    end for
    for d = 1 to m − 1 do
        for i = 1 to m − d do
            X(i, i + d) = ∅
            for s = i to i + d − 1 do
                for α in X(i, s) do
                    for β in X(s + 1, i + d) do
                        X(i, i + d) = X(i, i + d) ∪ (α, β)
                    end for
                end for
            end for
        end for
    end for
    if t ∈ X(1, m) then
        return Yes
    else
        return No
    end if
end procedure
```

Correctness:

By induction on recurrence. If $X = (i, j)$ correctly captures all possible symbols within set of $\{p_i, ..., p_j\}$, then by expanding the right border of the range by one increment and examing all possible cuts of the new range $\{p_i, ..., p_{j+1}\}$, all possible symbols in the new incremented range can also be captured in $X = (i, j + 1)$. Thus, the algorithm can capture all possible solutions of symbols from p after examining all possible segments of i, j.

Runtime Analysis:

Each step of inner two for loops takes $O(n)$. Each outer three for loops takes $O(m)$. Thus, the algorithm takes $O(n^2m^3)$ in total.