8-1.
Subproblems:
- d(i, j) ::= "The maximum profit filling the orders within from first order to i-th order using oil barrels not exceeding j."

Relate:
- $d(i, j) = \max \begin{cases} d(i-1, j) & \text{always} \\ d(i-1, j-a_i) + s \cdot a_i + p_i & \text{, if } j \geq a_i. \end{cases}$

Topological order:
- Both i and j depend on only lower integers.

Base case:
d(i, 0) = 0 for i = 1, ... , n
d(0, j) = sj for j = 1, ... , m

Original problem:
- d(n, m)
- Remember parent pointers to reconstruct the real problem.

Time analysis:
- The number of subprobs <= (n + 1)(m +1) = O(nm)
- O(1) per subproblem
- pseudopolynomial time

8-2.
Subproblems:
- x(i) ::= "The maximum value starting with pins i, i + 1, ... , n"
  y(i, j) ::= "The maximum value by hitting every pin i, i + 1, ... , j."

Relate:
- For cases: does not hit any pin, hit single pin, hit two adjacent pins, hit two pins which there is not any pin between them.

- $x(i) = \max \begin{cases} x(i+1) \\ v_i + x(i+1) \\ v_i \cdot v_{i+1} + x(i+2) \\ v_i \cdot v_k + y(i+1, j-1) + x(j) & \text{for } j \in \{i+2, \dots, n\} \end{cases}$

$y(i, j) = \max \begin{cases} v_i + y(i+1, j) \\ v_i \cdot v_{i+1} + y(i+2) \\ v_i \cdot v_k + y(i+1, k-1) + y(k+1, j) & \text{for } j \in \{i+2, \dots, j\} \end{cases}$

Topological order:
- decreasing i for x(i)
- decreasing j - i for y(i, j).

Base case: x(n + 1) = 0, y(i, j) = 0 if i > j

Original problem: x(1)

Time:
- O(n) sumproblems for x(i)
- For y(i, j), y(i', j') for i' $\in$ { i + 1, ... , n }, j' $\in$ { i', ... , n } was defined before.
- So O(n) subproblems for y(i, j).
- O(n)-time per single subproblem.
- O(n^3)-time
- polynomial time

8-3.

8-4.

First of all, compute f(w) for every word w appearing in L_n.
Let W denote a hash table mapping w to f(w). When let S denote a list of words appearing in L_q,
for w $\in$ S, if w in W, W[w] = W[w] + 1
else w not in W, W[w] = 1.
There are $\Omega(1)$ or $O(m)$ words in a log in L_u, so the algorithm takes $O(m^3n)$-time.

Subproblems:
- x(i, j) ::= "Maximized $\Sigma_{w \in R_{x,i,j}}$ f(w) where $R_{x,i,j}$ is a sequence of words restored from a substring of l_x ranging from i to j."
for x-th log in L_c.

Relate:
- For x(i, j), consider cases:
f(l_x[i : j])
x(i, k) + x(k + 1, j) for some k $\in \{i, \ldots, j\}$.
- x(i, j) = max $\begin{cases} f(l_x[i:j]) \\ x(i,k) + x(k+1, j) \text{ for } k \in \{i, \ldots, j\} \end{cases}$

Topological order:
- increasing x
- x(i, j) depends only on x(i', j') where j' - i' < j - i.

Base case:
- x(i, i) = W[l_x[i : i]] for i $\in \{1, \ldots, |l_x|\}$

Original problem:
- x(1, $|l_x|$) for x $\in \{1, \ldots, n\}$

- Store parent pointers to reconstruct the real question.

Time analysis:
- $O(m^3 n)$-time for preprocessing
- O(m^2) subprobs
- O(m)-time per subprob
- Compute $x(1, |l_x|)$ for n times.

$O(m^3 n) + (O(m^2) \cdot O(m)) \cdot O(n) = O(m^3 n)$ - time.