Mihai Dan
1/19/2016

Practice Problems #2

Problem 1:

n \ K	0	1	2	3
0	0	0	0	0
1	0	1/Y	1/N	1/N
2	0	1/N	2/Y	3/Y
3	0	1/N	2/N	3/Y

To populate the table, you add the numbers on the right, and you do not stop until you reach K.

$$\begin{cases} A[n] = A[n] + A[n-1] & \text{if less than } K \\ \text{otherwise}, & A[n] \end{cases}$$

Pseudocode

```
for i=1 to n
  A[i, j] = 0
for i=n to 1
  for x=n to i-x
    if (A[i, j] + next value) < K
      A[i, j] = A[i, j] + next value
  end
end
```

The runtime for this algorithm is $\Theta(n^2)$ because of the double nested for loop.

Problem 2:

n \ m	0	1	2	3
0	0	0	0	0
1	0	1/Y	1/N	1/Y
2	0	1/N	2/Y	3/Y
3	0	1/N	2/N	3/Y

To populate the table, you must use the algorithm below, which picks the highest profit for each scenario.

$$P[i] = \max \begin{cases} \max_{j \le i} [P[i] + (m_i, m_j) \cdot P_i] \\ P_i \end{cases}$$

Pseudocode:

```
for i=1 to n
    Profit[i]=0   set profit to 0
for i=2 to n
  for j=1 to i-1
     temp = Profit[j] * (m_i, m_j) · P[i]
     if temp > Profit[i]
         temp = Profit[i]
  if Profit[i] < P[i]
     Profit[i] = P[i]
```

The runtime for this algorithm is also $\Theta(n^2)$ because of the two nested for loops.

Problem 3:

i \ x	0	1	2	3
0	0	0	0	0
1	0	1/4	1/4	1/4
2	0	1/N	3/4	2/4
3	0	3/N	3/N	3/4

To populate the table, you must use the following algorithm to find the longest substring.

$$L(i,j) = \begin{cases} L(i+1, j-1) + 2 & \text{if } x[i] = x[j] \\ \max(L(i+1,j), L(i,j-1)) \end{cases}$$

$$L(i,j) = 1 \quad \forall i \in (1, \ldots, n)$$

Pseudocode:

```
for i=1 to n
   L[i,j]=1   used to store length of each subsequence
for i=1 to n
   for y=1 to n-i
      j = y+i
      L[y,j] = cost(L, x, y, j)
      L[j, y] = cost(L, x, j, y)
return L(1, n)
```

The runtime for this algorithm is $\Theta(n^2)$ because of the nested for loop.