Longest Palindrome Subsequence

We solve the longest palindrome subsequence (LPS) problem in a manner similar to how we compute the longest common subsequence in Section 15.4.

Step 1: Characterizing a longest palindrome subsequence

The LPS problem has an optimal-substructure property, where the subproblems correspond to pairs of indices, starting and ending, of the input sequence.

For a sequence $X = \langle x_1, x_2, \ldots, x_n \rangle$, we denote the subsequence starting at x_i and ending at x_j by $X_{ij} = \langle x_i, x_{i+1}, \ldots, x_j \rangle$.

***Theorem** (Optimal substructure of an LPS)*
Let $X = \langle x_1, x_2, \ldots, x_n \rangle$ be the input sequence, and let $Z = \langle z_1, z_2, \ldots, z_m \rangle$ be any LPS of X.

1. If $n = 1$, then $m = 1$ and $z_1 = x_1$.
2. If $n = 2$ and $x_1 = x_2$, then $m = 2$ and $z_1 = z_2 = x_1 = x_2$.
3. If $n = 2$ and $x_1 \neq x_2$, then $m = 1$ and z_1 is equal to either x_1 or x_n.
4. If $n > 2$ and $x_1 = x_n$, then $m > 2$, $z_1 = z_m = x_1 = x_n$, and $Z_{2,m-1}$ is an LPS of $X_{2,n-1}$.
5. If $n > 2$ and $x_1 \neq x_n$, then $z_1 \neq x_1$ implies that $Z_{1,m}$ is an LPS of $X_{2,n}$.
6. If $n > 2$ and $x_1 \neq x_n$, then $z_m \neq x_n$ implies that $Z_{1,m}$ is an LPS of $X_{1,n-1}$.

Proof Properties (1), (2), and (3) follow trivially from the definition of LPS.

(4) If $n > 2$ and $x_1 = x_n$, then we can choose x_1 and x_n as the ends of Z and at least one more element of X as part of Z. Thus, it follows that $m > 2$. If $z_1 \neq x_1$, then we could append $x_1 = x_n$ to the ends of Z to obtain a palindrome subsequence of X with length $m + 2$, contradicting the supposition that Z is a *longest* palindrome subsequence of X. Thus, we must have $z_1 = x_1$ $(= x_n = z_m)$. Now, $Z_{2,m-1}$ is a length-$(m - 2)$ palindrome subsequence of $X_{2,n-1}$. We wish to show that it is an LPS. Suppose for the purpose of contradiction that there exists a palindrome subsequence W of $X_{2,n-1}$ with length greater than $m - 2$. Then, appending $x_1 = x_n$ to the ends of W produces a palindrome subsequence of X whose length is greater than m, which is a contradiction.

(5) If $z_1 \neq x_1$, then Z is a palindrome subsequence of $X_{2,n}$. If there were a palindrome subsequence W of $X_{2,n}$ with length greater than m, then W would also be a palindrome subsequence of X, contradicting the assumption that Z is an LPS of X.

(6) The proof is symmetric to (2). ■

The way that the theorem characterizes longest palindrome subsequences tells us that an LPS of a sequence contains within it an LPS of a subsequence of the sequence. Thus, the LPS problem has an optimal-substructure property.

Longest Palindrome Subsequence

Step 2: A recursive solution

The theorem implies that we should examine either one or two subproblems when finding an LPS of $X = \langle x_1, x_2, \ldots, x_n \rangle$, depending on whether $x_1 = x_n$.

Let us define $p[i, j]$ to be the length of an LPS of the subsequence X_{ij}. If $i = j$, the LPS has length 1. If $j = i + 1$, then the LPS has length either 1 or 2, depending on whether $x_i = x_j$. The optimal substructure of the LPS problem gives the following recursive formula:

$$p[i, j] = \begin{cases} 1 & \text{if } i = j \ , \\ 2 & \text{if } j = i + 1 \text{ and } x_i = x_j \ , \\ 1 & \text{if } j = i + 1 \text{ and } x_i \neq x_j \ , \\ p[i+1, j-1] + 2 & \text{if } j > i + 1 \text{ and } x_i = x_j \ , \\ \max(p[i, j-1], p[i+1, j]) & \text{if } j > i + 1 \text{ and } x_i \neq x_j \ . \end{cases}$$

Longest Palindrome Subsequence

Step 3: Computing the length of an LPS

Procedure LONGEST-PALINDROME takes a sequence $X = \langle x_1, x_2, \ldots, x_n \rangle$ as input. The procedure fills cells $p[i,i]$, where $1 \leq i \leq n$, and $p[i,i+1]$, where $1 \leq i \leq n-1$, as the base cases. It then starts filling cells $p[i,j]$, where $j > i+1$. The procedure fills the p table row by row, starting with row $n-2$ and moving toward row 1. (Rows $n-1$ and n are already filled as part of the base cases.) Within each row, the procedure fills the entries from left to right. The procedure also maintains the table $b[1..n, 1..n]$ to help us construct an optimal solution. Intuitively, $b[i,j]$ points to the table entry corresponding to the optimal subproblem solution chosen when computing $p[i,j]$. The procedure returns the b and p tables; $p[1,n]$ contains the length of an LPS of X. The running time of LONGEST-PALINDROME is clearly $\Theta(n^2)$.

```
LONGEST-PALINDROME(X)
  n = X.length
  let b[1..n, 1..n] and p[0..n, 0..n] be new tables
  for i = 1 to n − 1
      p[i, i] = 1
      j = i + 1
      if x_i == x_j
          p[i, j] = 2
          b[i, j] = "↙"
      else p[i, j] = 1
          b[i, j] = "↓"
  p[n, n] = 1
  for i = n − 2 downto 1
      for j = i + 2 to n
          if x_i == x_j
              p[i, j] = p[i + 1, j − 1] + 2
              b[i, j] = "↙"
          elseif p[i + 1, j] ≥ p[i, j − 1]
              p[i, j] = p[i + 1, j]
              b[i, j] = "↓"
          else p[i, j] = p[i, j − 1]
              b[i, j] = "←"
  return p and b
```

Longest Palindrome Subsequence

Step 4: Constructing an LPS

The b table returned by LONGEST-PALINDROME enables us to quickly construct an LPS of $X = \langle x_1, x_2, \ldots, x_m \rangle$. We simply begin at $b[1, n]$ and trace through the table by following the arrows. Whenever we encounter a "$\swarrow$" in entry $b[i, j]$, it implies that $x_i = y_j$ are the first and last elements of the LPS that LONGEST-PALINDROME found. The following recursive procedure returns a sequence S that contains an LPS of X. The initial call is GENERATE-LPS$(b, X, 1, X.length, \langle\rangle)$, where $\langle\rangle$ denotes an empty sequence. Within the procedure, the symbol $||$ denotes concatenation of a symbol and a sequence.

GENERATE-LPS(b, X, i, j, S)

```
if i > j
    return S
elseif i == j
    return S || x_i
elseif b[i, j] == "↙"
    return x_i || GENERATE-LPS(b, X, i + 1, j − 1, S) || x_i
elseif b[i, j] == "↓"
    return GENERATE-LPS(b, X, i + 1, j, S)
else return GENERATE-LPS(b, X, i, j − 1, S)
```