Week 5 Notes

Monday, February 20, 2023 8:46 AM

Dynamic Programing

Change Problem
Finding the minimum number of coins for change

DPChange(*money*, *coins*)

$MinNumCoins(0) \leftarrow 0$
for m from 1 to $money$:
  $MinNumCoins(m) \leftarrow \infty$
  for i from 1 to $|coins|$:
    if $m \geq coin_i$:
      $NumCoins \leftarrow MinNumCoins(m - coin_i) + 1$
      if $NumCoins < MinNumCoins(m)$:
        $MinNumCoins(m) \leftarrow NumCoins$
return $MinNumCoins(money)$

"Programming" in "Dynamic Programming" has nothing to do with programming.

The Alignment Game

A T G T T A T A
A T C G T C C

Remove all symbols from two strings in such a way that the number of points is maximized

Remove the 1st symbol with both strings:
- 1 point if the symbols match
- 0 points if they don't match

Remove the 1st symbol from one of the strings
- 0 points

Example

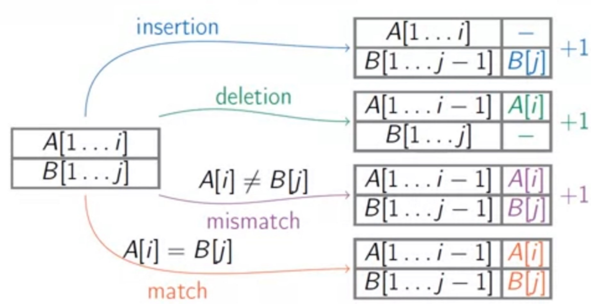
A T - G T T A T A
A T C G T - C - C
1 1   1 1   = 4

Alignment of two strings is two row matrix:
- 1st row: symbols of the 1st string (in order) interspersed by "-"
- 2nd row: symbols of the 2st string (in order) interspersed by "-"

Computing Edit Distance
Given string A[1…n] and B[1…m]. What is and optimal alignment of i-prefix A[1..i] of the first

string and a j-prefix B[1…j] of the second string

The last column of the optimal alignment is either an insertion, a deletion, a mismatch, or a match.
What is left is an optimal alignment of the corresponding tow prefixes.



$$D(i,j) = \min \begin{cases} D(i, j-1) + 1 & \\ D(i-1, j) + 1 & \\ D(i-1, j-1) + 1 & \text{if } A[i] \neq B[j] \\ D(i-1, j-1) & \text{if } A[i] = B[j] \end{cases}$$

Reconstructing an Optimal Alignment
The back tracking pointers that we stored will help us to reconstruct an optimal alignment

OutputAlignment(i, j)

if $i = 0$ and $j = 0$:
  return
if $backtrack(i, j) = \downarrow$:
  OutputAlignment$(i-1, j)$
  print $\begin{array}{|c|}\hline A[i] \\ \hline - \\ \hline \end{array}$
else if $backtrack(i, j) = \rightarrow$:
  OutputAlignment$(i, j-1)$
  print $\begin{array}{|c|}\hline - \\ \hline B[j] \\ \hline \end{array}$
else:
  OutputAlignment$(i-1, j-1)$
  print $\begin{array}{|c|}\hline A[i] \\ \hline B[j] \\ \hline \end{array}$