Backtracking part 2

1. A *subsequence* is obtained by deleting zero or more entries from a sequence without changing the order of the remaining entries. A *common subsequence* of two sequences is a subsequence of both. Given sequences $A[1..m]$ and $B[1..n]$, find the maximum possible length of a common subsequence.

   (a) Find a longest common subsequence of `ABC` and `BAC`. Is it unique? Explain why a common subsequence need not be a common substring.

   (b) Define $L(i, j)$ as the length of a longest common subsequence of the prefixes $A[1..i]$ and $B[1..j]$. Determine the base cases.

   (c) Suppose $A[i] \neq B[j]$. Explain why a common subsequence cannot use both final positions. Identify two smaller subproblems whose answers are sufficient to determine $L(i, j)$.

   (d) Suppose $A[i] = B[j]$. Explain why there exists a longest common subsequence that uses these two positions as a matched pair. Which smaller subproblem remains after removing that pair?

   (e) Combine your cases into a recurrence and then recursive pseudocode. State the initial call and identify a measure that decreases in every recursive call.

   (f) Draw the first few levels of the recursion for `ABC` and `BAC`. Circle repeated subproblems. What information would you save to avoid solving them again?