HW 1 Solution to the RNA matching count problem.

Problem 3. Now give recurrences that can be used to correctly count the number of permitted matchings, and give a DP that computes the count in \(O(n^3)\) time. This may be a bit challenging.

Solution: \(S(i, j)\) is defined as the number of permitted matchings involving the positions from \(i\) to \(j\) inclusive. It includes the empty matching as one of the matchings. For technical reasons, define \(S(j, j) = S(j + 1, j) = 1.\)

Then
\[
S(i, j) = S(i + 1, j - 1) + \sum_{i < k < j; \text{ } i \text{ and } k \text{ can match}} [S(i + 1, k - 1) \times S(k + 1, j)] + \sum_{i < k < j; \text{ } j \text{ and } k \text{ can match}} [S(i + 1, k - 1) \times S(k + 1, j - 1)]
\]

The first term covers the case that neither \(i\) nor \(j\) are involved in a match. The first sum counts all of the cases where \(i\) is involved in a match to some position between \(i + 1\) and \(j\). Even though \(i\) is definitely matched to a position \(k\), \(j\) may or may not be involved in a match, and the number of those possibilities is \(S(k + 1, j)\). The second sum counts the cases where \(j\) is definitely involved in a match and \(i\) definitely is not involved. We need to verify that all possible matchings are counted, and that none are double counted. The cases are: 1) that neither \(i\) nor \(j\) are involved in a match - that is covered in the first term, but not in either of the sums, since they count only matchings where \(i\) or \(j\) or both are in a match.

2) \(i\) definately is in a match. Those matchings are counted in the first sum, but not in the second sum or the first term.

3) \(i\) and \(j\) are both in a match. Those matchings are counted in the first sum, but not in the second sum or the first term.

4) \(j\) is in a match, but \(i\) is not. Those matchings are counted in the second sum.