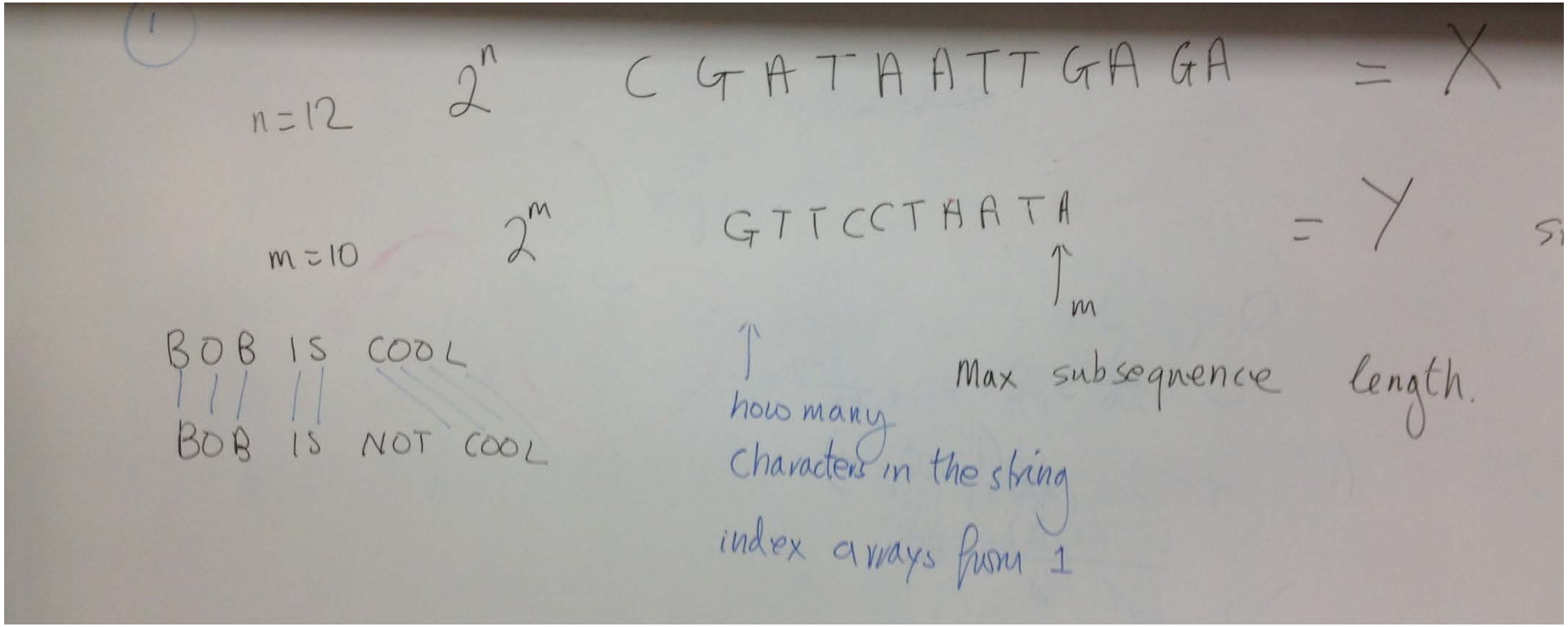
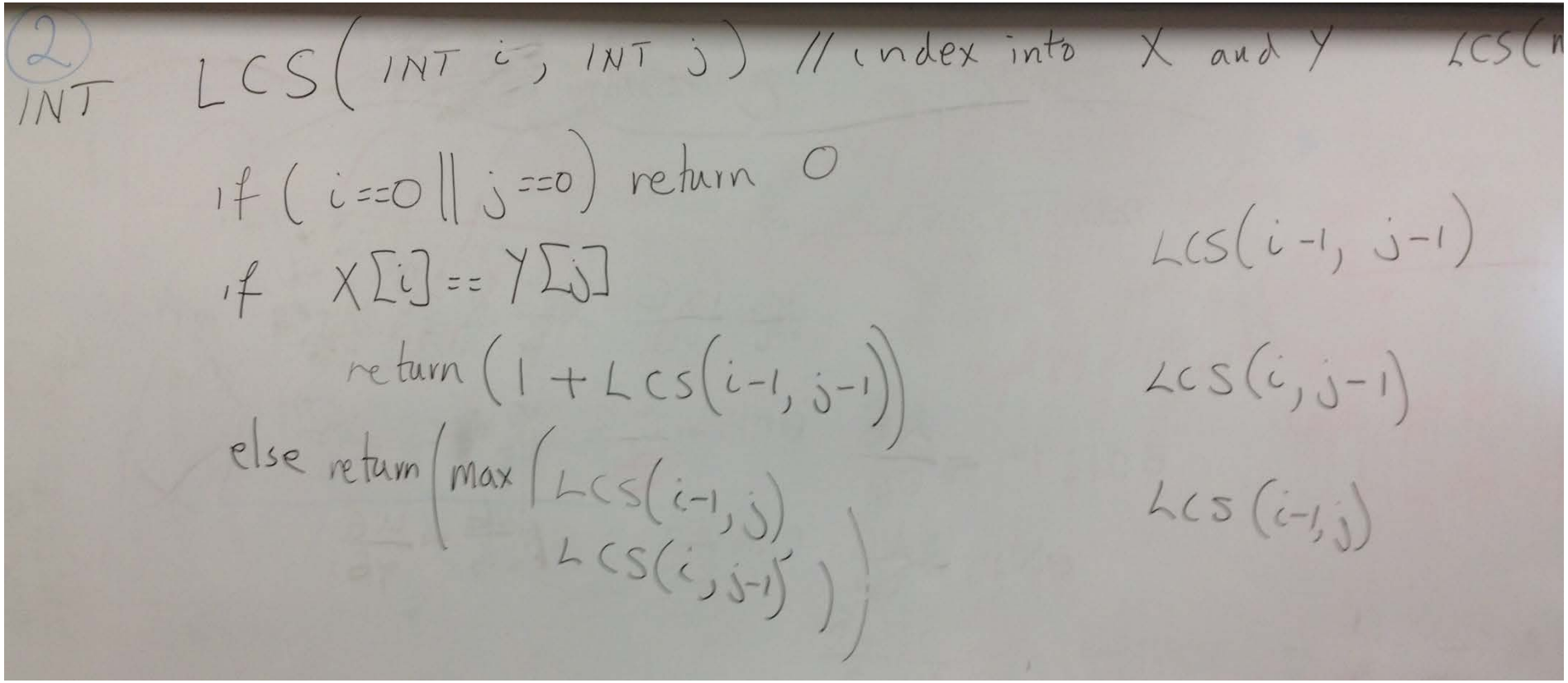
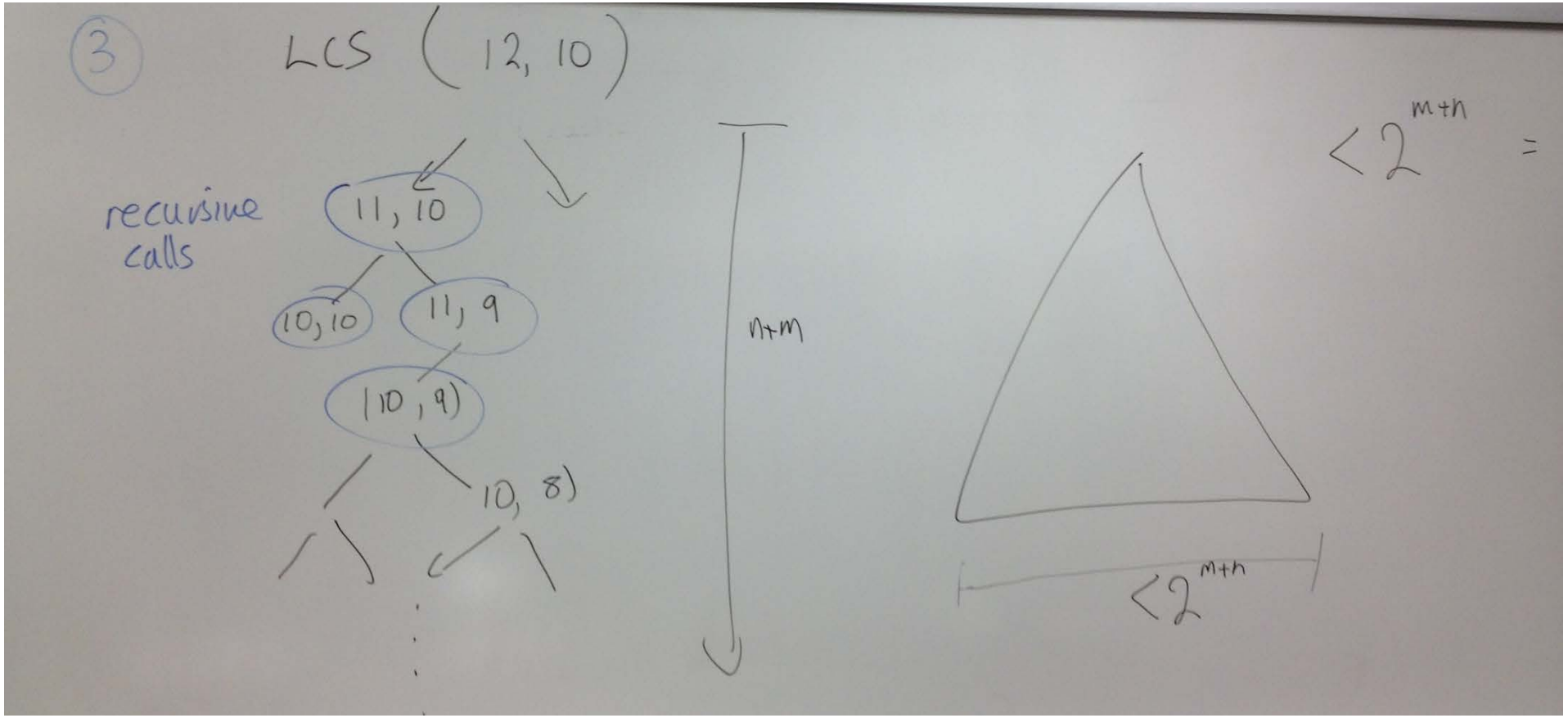
Cs5050 notes 01 21 14



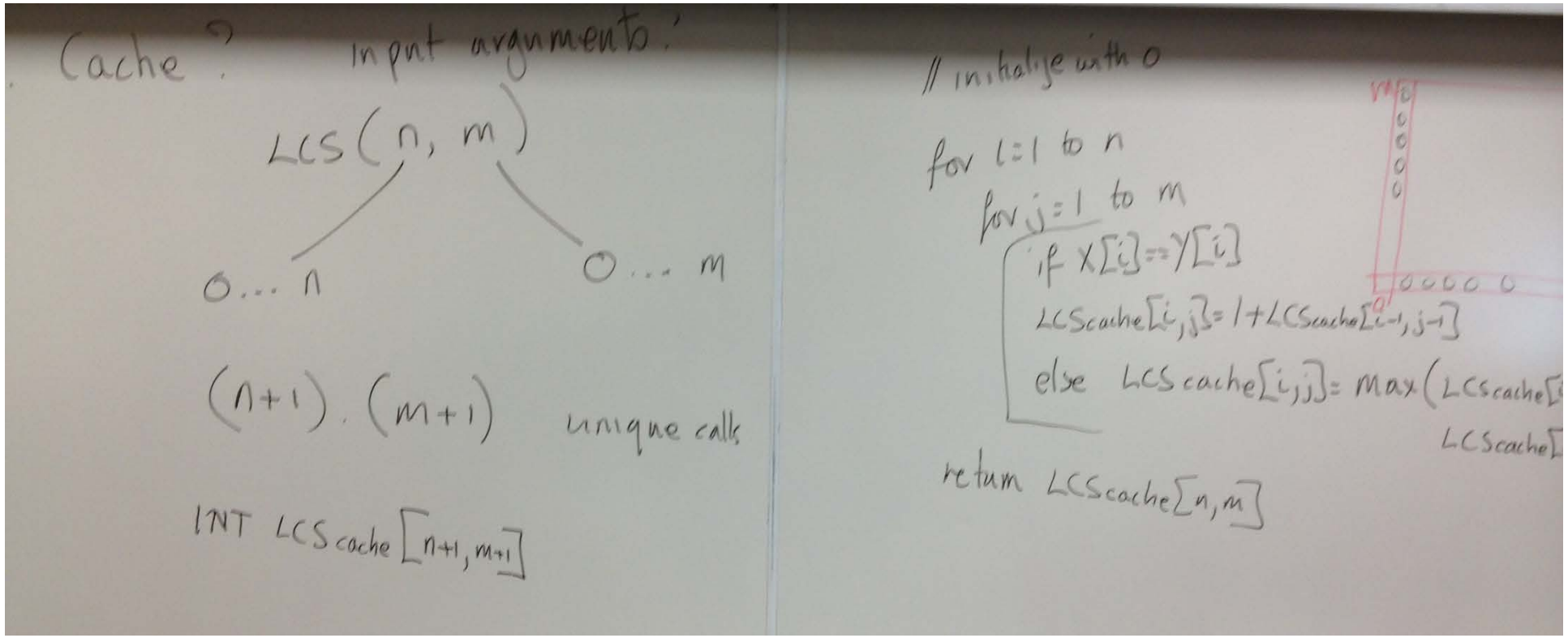
Longest common subsequence problem.
A subsequence of a string A is a string contains a subset of characters from A preserving the order
So for the top string above, CGAA is a subsequence, but AGT is not. For a string of length n, there are 2^n substrings
Given two strings find the longest subsequence that exists in both strings
Brut force algorithm tries every combination takes $2^{(n+m)}$ steps



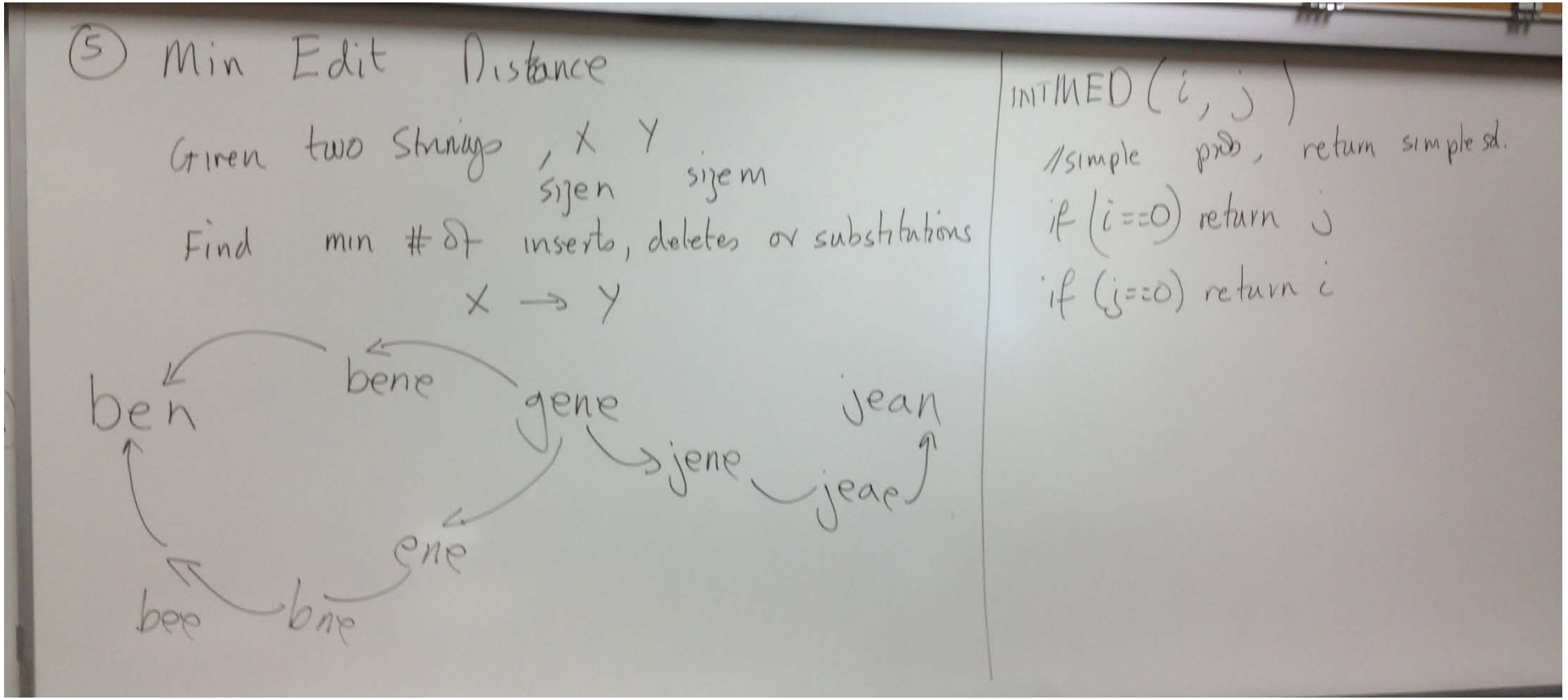
Using the meta algorithm technique creates a simple algorithm to solve the problem.
This algorithm returns the length of the LCS, not the sequence itself.



Visualizing the calling tree we get a tree with a branching factor of 2 and a depth between min(m,n) and m+n
Hence, the algorithm will take around $2^{(n+m)}$ steps



Recall that the arguments to the function represent the possible problem instances.
So since the first argument is the index into X and the second the index into Y, then there are only (n+1)*(m+1) Unique function calls.
So we can create a simple DP algorithm that scans through an integer 2D array



Next problem is the minimum edit distance with applications in spell checking and Bioinformatics