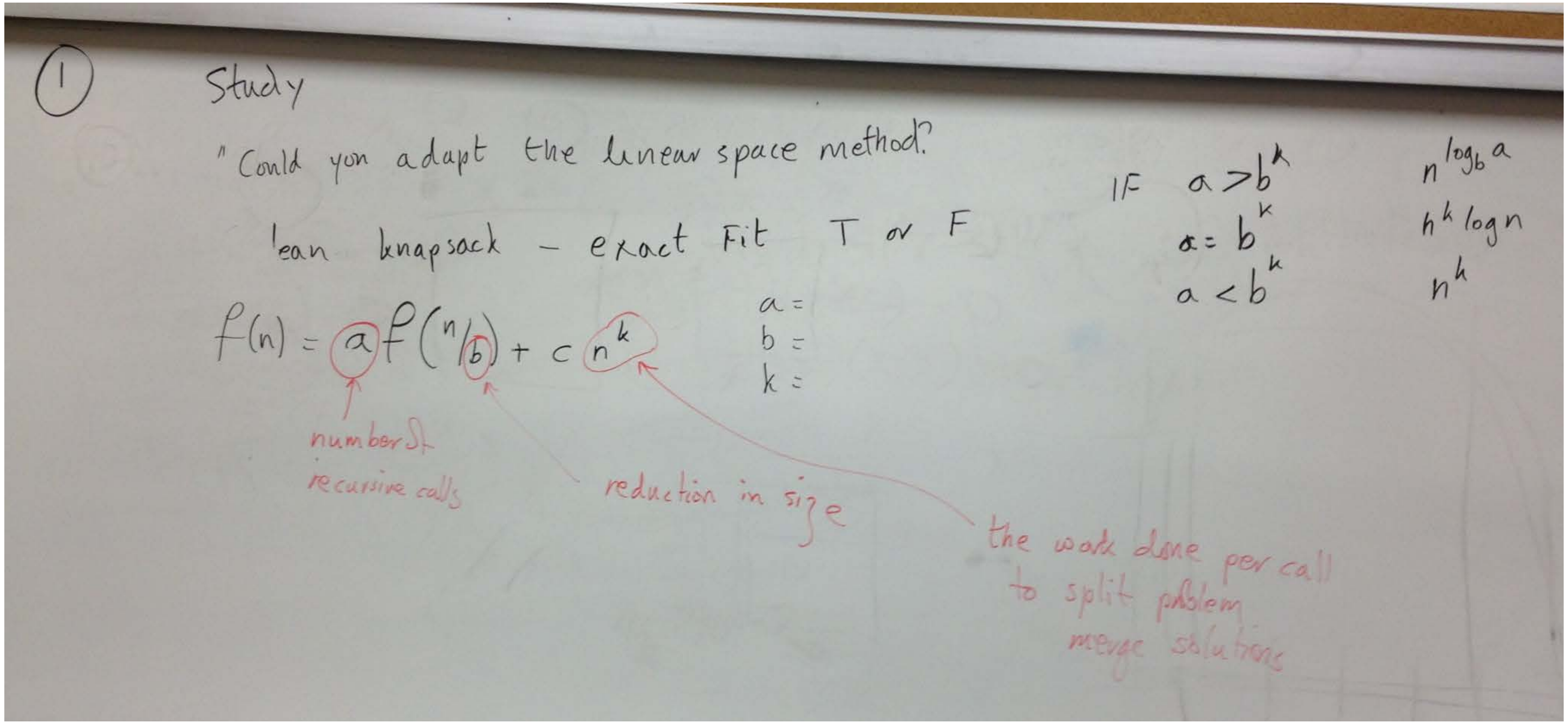
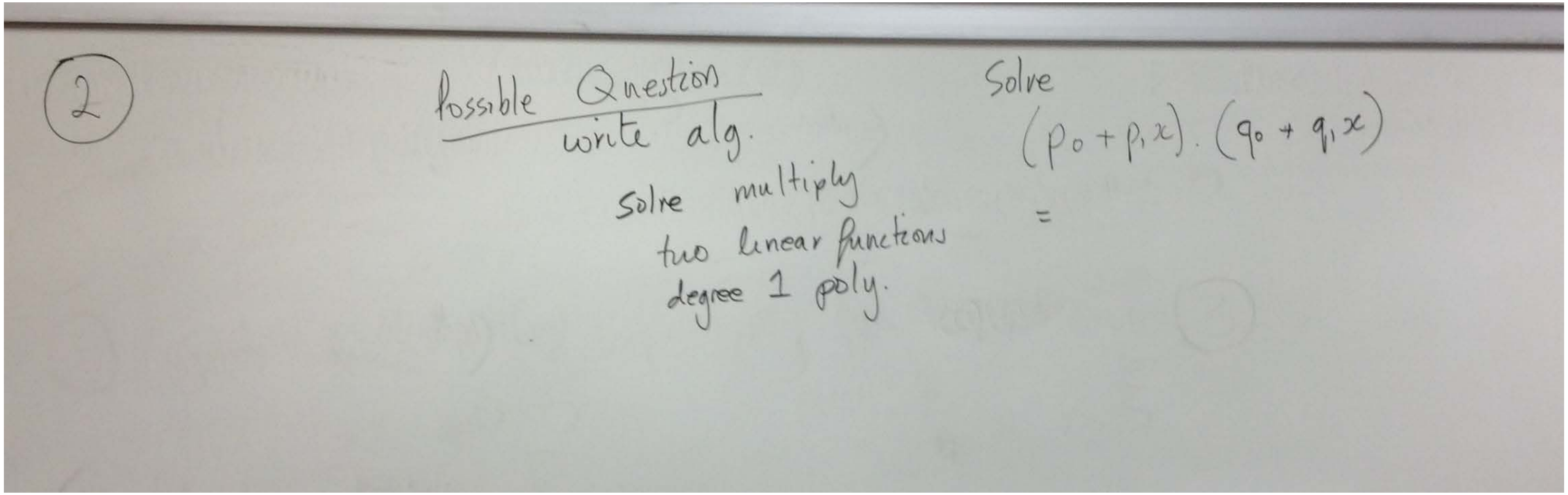
cs5050

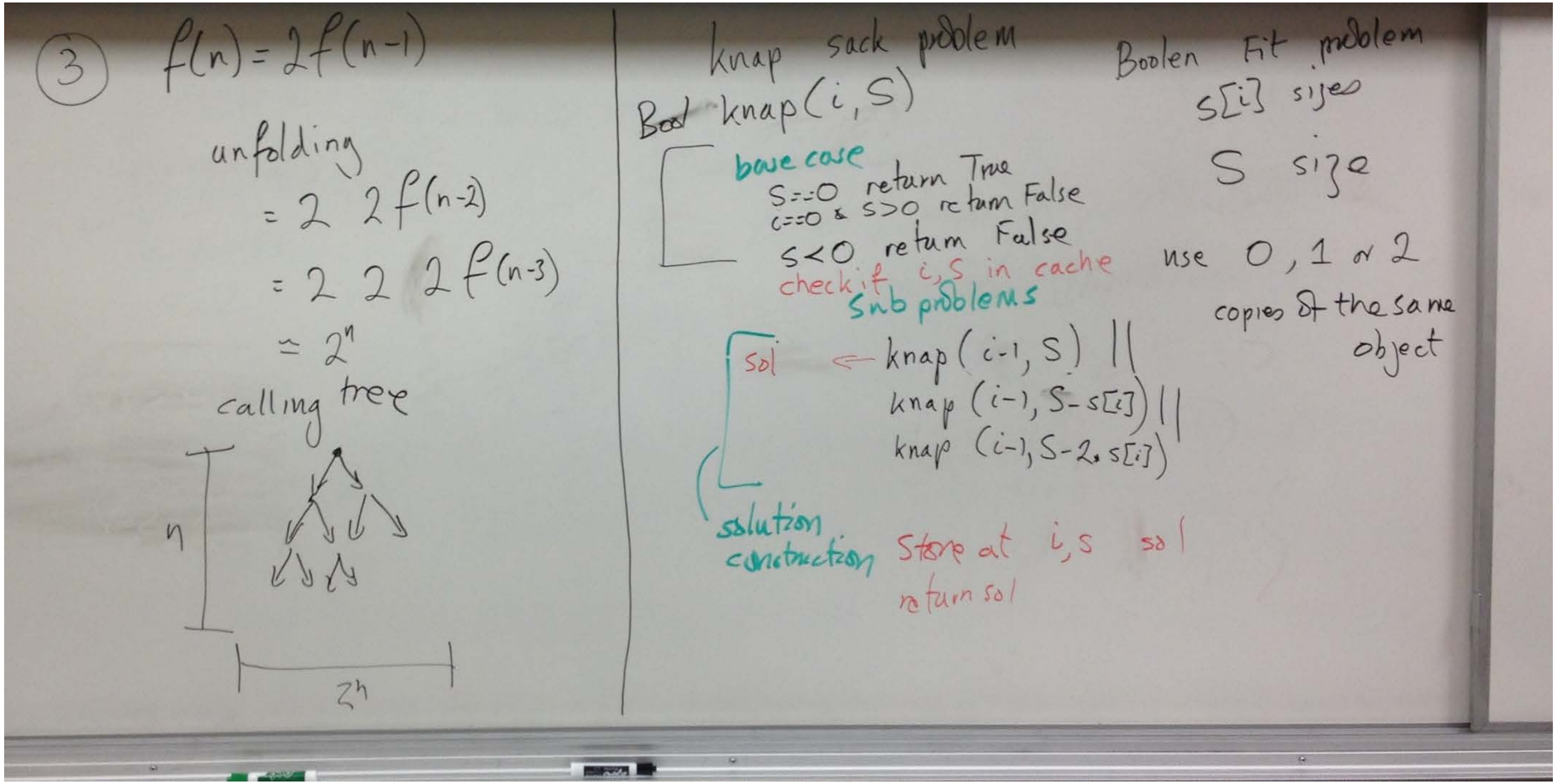
02_20_14



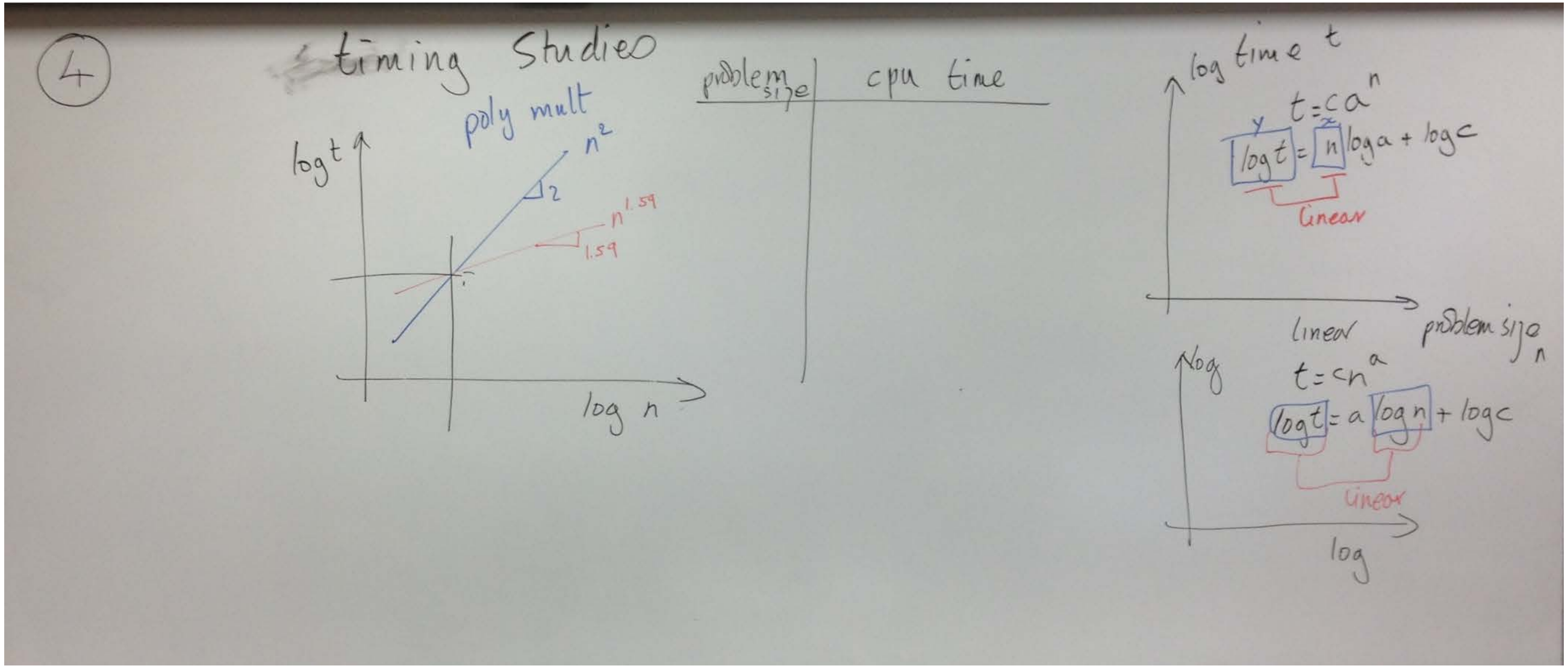
1) Understand what the parameters a, b and c mean



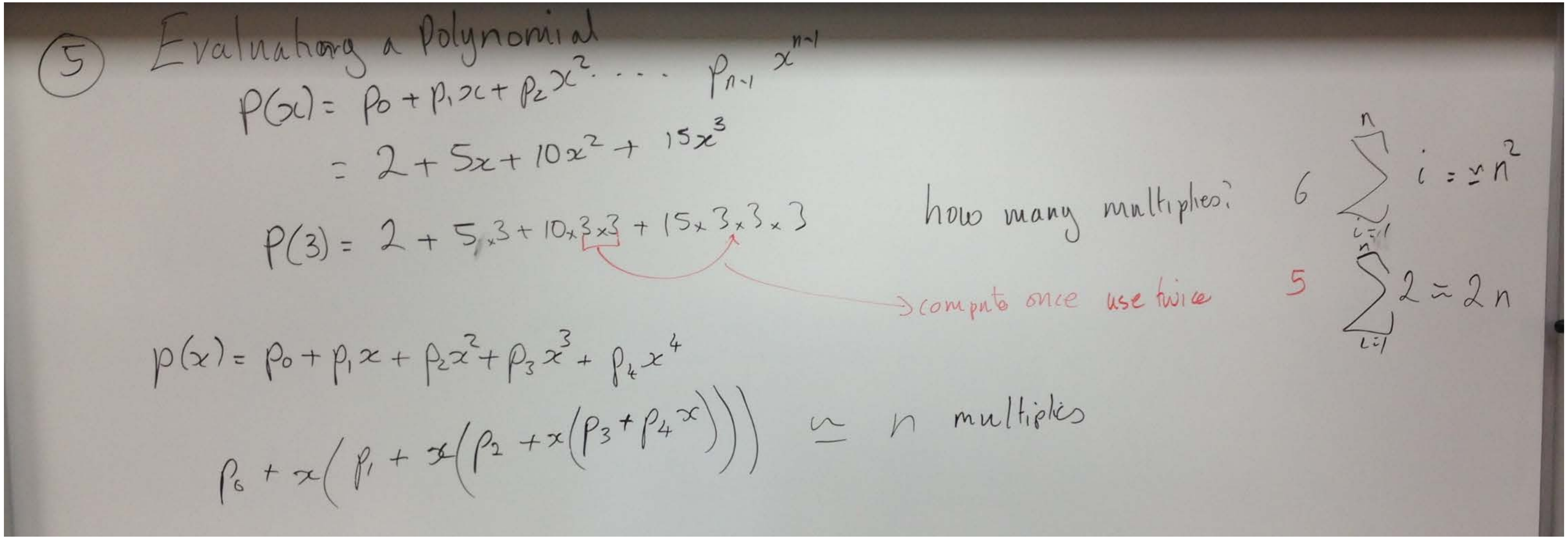
1) Is there an efficient way to multiply two linear functions to determine the coefficients of the resulting quadratic?



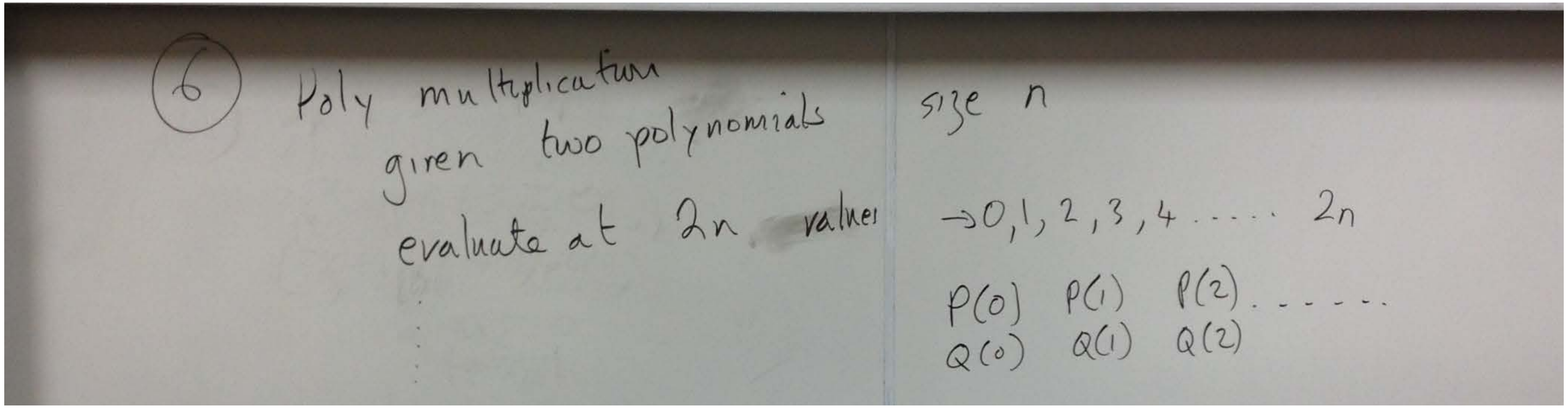
1) Given a problem description, can you create a recursive solution? Can you then analyze it to determine the number of recursive calls it will generate? Can you identify the cache size needed? Can you create a DP algorithm from the recursive definition?



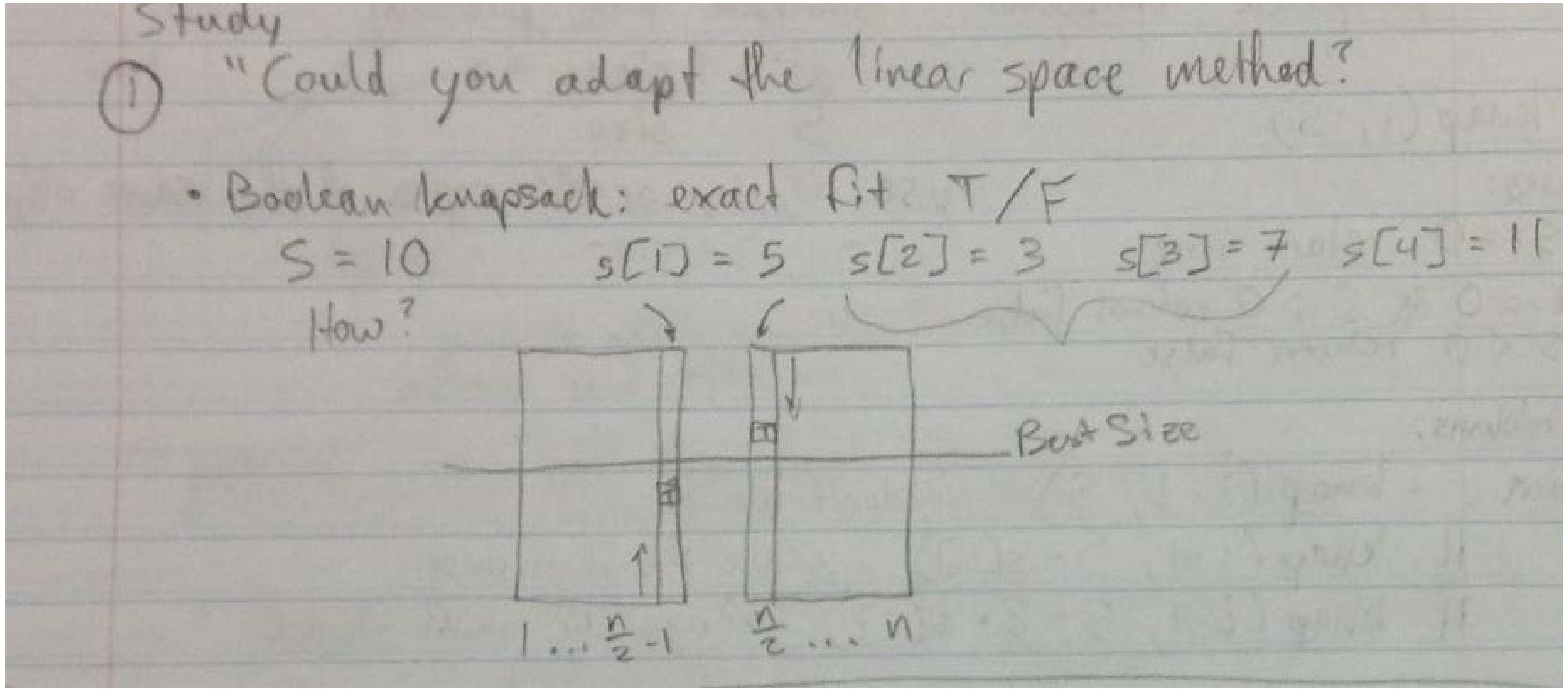
1) Use a linear log plot when the expected performance is **ca^n**
2) Use a log log plot when the expected performance is **cn^a**
3) Why?
4) What is the significance of the slope and offset of the lines on the graph?
5) What is the significance of the crossing point of two algorithm performances?



1) Different ways to evaluate a polynomial of degree n
    1) Simple way where $p_i x^i$ takes i+1 multiples, needs a total of n^2 multiplies
    2) Caching x^i to compute x^{i+1} takes multiples total 2n
    3) Using the factoring technique above (add then multiply) takes only n total



1) New way of multiplying two polynomials P and Q with n terms each to find PQ, the polynomial that is P times Q
    1) First evaluate each at 2n values (why do we need this many?, because the result has 2n-1 terms we need at least this many to find the unknowns. If we have 2n unknowns, we need at least 2n equations)
    2) Multiply each P(j) by Q(j) to get PQ(j), where j is the value of x. We know that PQ must pass through each y=PQ(i) point. We interpolate to solve for the coefficients given points on the polynomial (we have not done this step yet, so don't bother understanding it now).



1) You know how to use the linear space, D&C algorithm to find the objects in the max value knapsack problem. Can you adapt the method to find the objects in a "find a subset of the objects that exactly fit into a knapsack" Boolean problem?