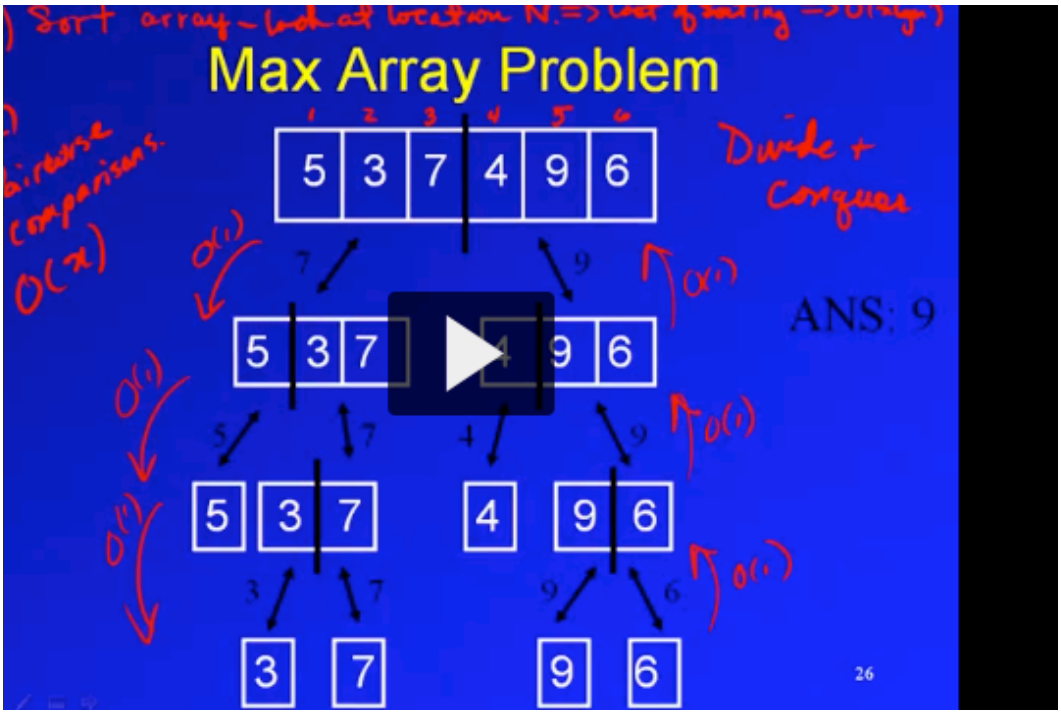
In the lecture material for this module, you are asked to consider submitting a solution for the cost of the Max Array problem. This is entirely extra credit. If you wish to submit, please do so by Tuesday midnight, by clicking on the "Max Array Problem" link above and submitting an attachment. I suggest you start by drawing pictures.

So the cost of dividing each subarray is constant time from one level to the next.... we just have to calculate the midpoint and split the group. Passing up the winners from each group is also a constant time cost. This is so far very similar to binary search but more work than binary search, but hard to tell by how much. It's more work than binary search... which is log(n) so that's the lower bound of max array.

Max array almost seems to be doing 5 divisions for an element **6 sized array.** Maybe the cost is n which is > logn?



TYPE TO ENTER A CAPTION.

For an 8 array element, you'd divide them into 4,4 (1st divide) and the left side would be divided into 2,2 (second divide), and the left side is divided into 1,1 (third divide) and the right side is 1,1 (fourth divide). So each 4 has 3 divides so the 4,4 has a total of 6 divides + 1 more for the initial divide into 4,4. That's 7 divides. It seems like it grows linearly with n?

Maybe the cost is O(n)