

COP 3502 Suggested Problems/Program Edits: Recurrence Relations, n² Sorts (Week 8)

1) Use the iteration technique to determine a Big-Oh solution for the following recurrence relation:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2, T(1) = 1$$

2) Solve the following recurrence relation defined for non-negative integers, n, using the iteration technique. Please solve the recurrence **exactly**, obtaining a closed-form solution for T(n), in terms of n.

$$T(n) = 2T(n - 1) + 2^n, \text{ for } n > 0 \\ T(0) = 1$$

3) Edit Bubble Sort so that if a whole iteration completes without making any swaps, then you don't run any more iterations and return.

4) Rewrite insertion sort so that the sorted list is built from the right side and not the left. So, for example, if the array first has:

3 2 8 1 6

It would look like this after each iteration:

3	2	8	1	6	(inserted 1 into list with 6)
3	2	1	6	8	(inserted 8 into list 1,6)
3	1	2	6	8	(inserted 2 into list 1,6,8)
1	2	3	6	8	(inserted 3 into list 1,2,6,8)

5) In selection sort the way it was taught in class, we place the maximum element of the array at the end first. Reverse the logic so that the first item that is selected is the smallest, the second item selected is the second smallest, etc.