**CS430 Lecture 04 Activities**

1. A recurrence relation describes runtime function recursively for a recursive algorithm. Write a recurrence relation for the Mergesort algorithm. HINT: try to count the number of executions of each statement and the cost of each statement.

Solving Recurrence Relations – Recurrence Tree Method

We solve a recurrence relation to get a function in its closed (non-recursive) form. The recurrence tree method is a visual method of repeatedly substituting in the recurrence relation for T(n) on smaller and smaller “n” until you reach the base case, and then summing up all the nodes in the tree.

2. Draw the recurrence tree for Mergesort

Divide and Conquer Algorithms

* Divide – divide the problem into sub-problems that can be solved independently
* Conquer – recursively solve each sub-problem
* Combine – possibly necessary, combine solutions to sub-problems

Not all problems can be solved with the divide and conquer approach. Maybe sub-problems are not independent, or solutions to sub-problems cannot be combined to find solution to main problem.

3. Write a recursive algorithm for Binary Search. Write and solve its recurrence relation.

4. Write a recursive algorithm for Selection Sort (or insertion sort or bubble sort). Write and solve its recurrence relation.

5. Describe an efficient divide and conquer algorithm to count the number of times a character appears in a string of length n.

Inductive Proofs (needed in next lecture to prove a solution to a recurrence relation)

6. What are the three steps in an inductive proof?

7. Use an inductive proof to show the sum of the first n integers is (n)(n+1)/2