1. Material covered (or assumed covered) so far:

(a) Growth of functions
(b) Conventions in writing pseudocode
(c) Analysis techniques for algorithms
(d) Solving recurrences
(e) Summing series
(f) Searching algorithms (linear and binary)
(g) Insertion Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort.
(h) Lower bound on comparison based sorting algorithms.
(i) Divide-and-conquer design paradigm for algorithms
(j) Linear time (non-comparison based sorting algorithms)-counting sort and radix sort
(k) Max and Min of a set of elements. Lower bound for simultaneous max and min.
(l) General selection problems. Finding the rank of an element (relatively easy and simple worst case linear time algorithm).
(m) Finding an element of a given rank (in particular the median). Expected linear time algorithm, and worst case linear time algorithm.
(n) Probabilistic analysis of randomised algorithms (specifically randomised quick sort and randomised select).

(o) The material for these are from chapters 3-9 in the third edition of the textbook by Cormen, Lieserson, Rivest and Stein.

(p) Dynamic programming: The general theory and the specific examples of Fibonacci numbers, profitable rod cutting, matrix chain multiplication and longest common subsequences as working examples. (the material for this is from chapter 15 of the text book).
2. To be covered from now:

(a) Greedy algorithms. The relationship to dynamic programming algorithms. The similarities and differences. Illustrative examples: Huffman codes, activity selection (special case of the famous graph colouring problem and more specifically its restriction to interval graphs), and the 0-1 knapsack problem (dynamic programming) versus the fractional knapsack problem (greedy algorithm).

(b) This material is from Chapter 16 of the textbook. Will be covered roughly during the lectures on Oct 22 and Oct 25.

(c) Graph Algorithms. Basic introduction. Specific problems:

(d) Fundamental graph searching algorithms: Bread First Search (BFS), Depth first search (DFS). Chapter 22.

(e) Minimum spanning trees (example of greedy algorithms) chapter 23

(f) Single source shortest paths. (example of greedy algorithms) chapter 24

(g) All-pairs shortest paths (example of dynamic programming). chapter 25. These will be covered beginning Oct. 28 till the lectures in the 2 weeks in November.