Course Overview

(Red items were not on the midterm.)

- $\Rightarrow$ Average case analysis:
  - Given an algorithm, find the worst case, average case, best case: we did ListSearch, quicksort, binary2decimal, leafs
  - How does randomization affect the expected worst case? - randomized quicksort
  - Expected value of properties of data structures, such as 2,3,4-trees, heaps, etc

- Dictionaries:
  - $\Rightarrow$ Hashing or Direct Addressing:
    - closed addressing–load factor, examples of hashing, chaining
    - open addressing–probe sequences, dynamic storage, nonlinear probing, linear probing, double-hashing
    - dynamic hashing: incremental, extensible
  - balanced search trees
  - 2,3,4-trees–operations–underflow, overflow and merging, borrowing, rotating, complexity
  - red-black trees
  - B-trees

- Augmented Data-structures, examples: rank, average, min for 2,3,4 trees, how to do all the operations we did before with these augmented data structures

- $\Rightarrow$ Amortized analysis: accounting, aggregate, binary counter, stack, dynamic arrays, disjoint sets

- Graph Theory
  - Searches: BFS, DFS,
  - BFS/DFS produce spanning trees of unweighted graphs and give path information, topological ordering
  - Spanning Trees of weighted graphs–Kruskal, Prim
  - Priority Queues $\Rightarrow$ heaps, Heapify, heapsort, insertion, deletion, extractMin, use an array to implement, complexity, heap implementation, direct addressing implementation
  - $\Rightarrow$ Disjoint Sets: all the implementations, complexity of them, how does adding union-by-weight help?, union-by-rank?, path-compression?