dont know	know a bit	OK	good!	master	COMS 311 TOPICS				
1	2	3	4	5	**BIG-OH**	**week4**	week3	week2	week1
					Basics				x
					Definitions of big-oh, omega, theta c>0, n>=0 (7 things)				x
					big-omega				x
					big-theta				x
					Intuition/understanding (graph)				x
					tighter and weaker bounds			x	x
					how to prove O/Omega/Theta techniques				x
					for polynomials choose c >sum of coeff or coeff for omega				x
					for same type compare exponents				x
					take log				x
					Application to Algorithms				x
					ram model (vs actual)				x
					instances and runtime graphs			x	x
					WCET, BCET, ACET			x	x
					Big-oh of code segments			x	x
					problem complexity and algorithmic complexity				x
					code examples of different Os				x
					Big-oh in real world			x	
					real code times (matrix mult)			x	
					effect of cache/pipelining etc			x	
					choosing algo in real-world vs big-Oh			x	
					constants might matter more in real world than O			x	
					easier implementation might make the diff			x	
					Dominance Relationships				x
					logs beat constants				x
					poly beats all logs				x
					exp beats all poly				x
					fact beats all exp				x
					n^n beats fact				x
1	2	3	4	5	**DATA STRUCTURES**				
					Basic			x	x
					arrays (sorted/unsorted)			x	x
					linked lists (singly/doubly; sorted/unsorted)			x	x
					comparison of arrays and linked lists			x	x
					Basic Abstract Data Types			x	x
					Stack, Queue (implementations using array/linkedlists)			x	x
					comparison of operations of diff impl of stack/queue			x	x
					reasons for differences			x	x
					Dictionaries		x	x	
					Hash Tables		x	x	
					Other ADTs and their Java Implementations		x	x	

						Binary Search Trees		x	x	
						Priority Search Queues		x	x	
						Heap impl		x	x	
						Fast Heap Impl and Analysis		x		
					Storing points, graphs, sets etc			x	x	
						graphs	x	x	x	
						sets	x	x	x	
					big-oh of operations on data structures			x	x	
					algorithms on data structures (BST, HEAP etc)			x	x	
1	2	3	4	5	**GRAPHS (unweighted)**					
					BFS		x	x		
						connected components	x	x		
						two-color problem	x	x		
					DFS on undirected graphs (tree and back edges)		x	x		
						articulation vertices (parent, root, bridge cutnodes)	x			
					DFS on directed graphs(tree,back,cross,forward)		x			
						topological sorting/DAG	x			
						strongly connected components	x			
						following algorithm by marking graph	x			
					big-oh of graph algorithms		x	x		
1	2	3	4	5	**P-NP**					
					Intro Concepts					x
						the diagram and four classes of problems	x			x
						informal (solvable, probab intract, provably intract, prov unsolvable)	x			x
						examples of problems in four classes	x			x
						Halting Problem	x			x
						Hamiltonian Cycle Enumeration problem	x			x
						Hamiltonian Cycle Search problem	x			x
						Independent Set problem	x			x
						Search/Sort problems	x			x
					Classes of problems					
						Undecidable (prove Halting problem is undecidable)	x			x
						P	x			x
						NP	x			x
						prove P is a subset of NP				
						NP-Complete (probably intractable class)				x
						why if a NP-C problem is in P, then P=NP				
						provably intractable classes				x
						NP-Hard problems				
					Reduction					
						optimization, search, decision and reductions	x			
						definition of polynomial-time reduction & notation	x			
						conclusions (or implications of) from Y <=p X	x			
						transitive property (x <=p y, y <=p z => x<=p z)	x			

						how to prove X <=p Y (the three steps)	x			
						interval sched <=p IS	x			
						vector cover <=p IS	x			
						IS <=p vector cover	x			
						bipartite matching <=p IS	x			
					NP-Complete					
						Definition and Intuition				
						The first NP-C problem (circuitSAT) & Cook-Levin's theorem				
						how to prove that a problem is NP-C				
						prove 3-SAT is NP-C				
						prove IS is NP-C				
						prove VC is NP-C				
1	2	3	4	5	**PROOF TECHNIQUES**					
					Why is proving important?					x
						Job selection problem				x
						solutions to JS problem				x
						why is proving so important?				x
					CALCULUS					PreReq
						propositional logic rules				PreReq
						predicate logic rules				PreReq
						form of deduction proofs				PreReq
					DIFFERENT TECHNIQUES				x	PreReq
						contradiction (and correct form)		x	x	PreReq
						induction (form)			x	PreReq
						direct proof (see form of deduction proofs)			x	PreReq
						trivial/vacuous			x	PreReq
						contrapositive	x		x	PreReq
					EXAMPLES IN CLASS					PreReq
						contradiction (and correct form)		x		PreReq
						induction (form)				PreReq
						direct proof (see form of deduction proofs)				PreReq
						trivial/vacuous				PreReq
						contrapositive	x			PreReq
					Proofs in class					
						Halting problem is undecidable		x	x	PreReq
						VC <=p IS and IS <=p VC	x			
						select jobs satisfies greedy choice + opt substructuring				
					Proving Code correct					
						Loop invariants				
						proof of recursive codes..				
1	2	3	4	5	**ALGORITHMIC TECHNIQUES**					
					BRUTE FORCE TECHNIQUES					

						search space for different problems			x	x
						recursion tree for brute force approach			x	x
						back track algo from text book			x	x
						iterative way to generate all subsets				
						recursive way to generate all subsets			x	x
						recursive way to generate all perms			x	x
						recursive way to gen size k subsets			x	x
					DIVIDE AND CONQUER					
						recurrence formula	x	x		
						divide and conquer approach	x	x		
						mergesort + analysis	x	x		
						quicksort + analysis	x	x		
						max	x	x		
						max sum of sequence	x			
						counting inversions	x			
						finding sink in graph				
					Recurrence Formulae		x	x		
						general form of recurrence formula & masters theorem	x	x		
						how to derive recurrence tree, term for each level, sum	x	x		
						2T(n/2) + c	x	x		
						T(n/2) + cn	x	x		
						2T(n/2) + cn	x	x		
						3T(n/2) + cn	x	x		
						2T(n/2) + cn^2	x	x		
						5T(n/2) + cn^2	x	x		
					GREEDY TECHNIQUE					
						greedy approach				
						interval scheduling by greedy approach				
						greedy choice property				
						optimal substructuring property				
						proving gc property				
						proving os property				