Major Algorithms

Algorithm/Framework	**Problems**	**Source**
Selection Sort	Sorting	Lecture 8
Insertion Sort	Sorting	Lecture 8
Prim's Algorihm***	Minimum Spanning Tree	Lecture 10
Greedy*	Knapsack, Coin Exchange	Lecture 10
Brute Force Bitlists*	Knapsack	Lecture 11
*Brute Force Permutations**	Traveling Salesperson, N-Queens	Lecture 11, 15
Binary Search	Search (in sorted list)	Lecture 12
Merge Sort	Sorting	Lecture 12
Quick Sort	Sorting	Workshops
Backtracking*	N-queens, Hamiltonian Cycles, Knapsack	Lecture 15
Depth-first-search (iter, recur)	Reachability, Height of Binary Tree**	Lectures 16, 18
Breadth-first-search	Reachability, Shortest Path (unit weights	Lecture 16
Dijkstra's Algorithm	Shortest Path (weighted)	Lecture 17
Binary Search Tree Search	Search (in BST)	Lecture 18
Binary Search Tree Insertion	Insert (into BST)	Lecture 18, Workshop
Heap Extract Min	Find and Remove Minimum (in heap)	Lecture 19, Workshop
Heap Insertion	Insert (into heap)	Lecture 19
Heap Sort	Sorting	Lecture 20

*Frameworks with different instantiations for concrete problems
**We did not refer to it as DFS but it technically is
***This is also an instantiation of the greedy algorithm but famous enough to go by it's own name

These are major non-trivial algorithms covered in the lecture that are important to know.
Algorithms in italic should be known in principle but student's are not expected to be able to reconstruct them in detail (under exam conditions).
Some smaller algorithms (e.g., Sequential Search, Finding Minimum in List, etc.) are ommited here but due to their simplicity one should certainly be able to reconstruct them under exam conditions.