15-110 Quiz4 Notes Sheet

Runtime and Big-O Notation

Best case: an input that leads to an algorithm taking the least steps possible
Worst case: an input that leads to an algorithm taking the most steps possible

Function family: a set of functions that all grow at a similar rate (eg, linear functions) expressed in a simplified format.
Common function families: constant, logarithmic, linear, quadratic, exponential

Big-O: a representation of the function family of the worst-case scenario for a specific algorithm. Represented as O(runtime).
Common Big-O runtimes: O(1), O(log n), O(n), $O(n^2)$, $O(2^n)$

Linear Search: O(n)
Binary Search: O(log n)
Hashed Search: O(1)

Trees

Tree: a data structure composed of nodes holding values that are connected hierarchically in a recursive manner.
Binary Tree: a tree where each node has at most two children, called left and right

Parent: the node connected directly above the current node
Children: the nodes connected directly below the current node
Root: the topmost node of a tree (with no parent)
Leaf: a node with no children

Our tree format is a recursively nested dictionary:

```
{ "contents" : nodeValue,
  "left"     : leftChildSubtree,
  "right"    : rightChildSubtree }
```

If there is no left/right child, the key maps to `None` instead.

The common algorithm structure for trees is recursive:
Base case: when the tree is a leaf, or an empty tree
Recursive Case: recursively call the function on the left child and right child (if they exist) and combine the results with the current node

Graphs

Graph: a data structure composed of nodes holding values connected by edges
Neighbors: a pair of nodes connected by an edge.
Directed: a graph where edges can go from one node to another and not vice versa. Opposite is undirected.
Weighted: a graph where edges have values (called weights). Opposite is unweighted.

Our graph format is a dictionary mapping nodes to lists of neighbors:

```
{ nodeValue : [ neighborValue ],
  ... }
```

If the graph is weighted, neighbors are represented as value-weight pairs:

```
{ node : [ [neighborValue, weight] ],
  ... }
```

15-110 Quiz4 Notes Sheet

Search Algorithms II

Binary search tree (BST): a tree for which the left child of every node (and all its children, etc) are strictly less than the node, and the right child of every node (and its children, etc) are strictly greater than the node.
Binary search: an algorithm that can be performed on a BST by making just one recursive call - to the left if the target is smaller than the root, to the right if larger.
Binary search runtime: binary search on a BST runs in O(log n) if the tree is balanced (all left and right subtree pairs are ~ the same size), O(n) if unbalanced.

Breadth-First Search (BFS): an algorithm where you search for a path from a start node to a target by visiting the immediate neighbors, then their immediate neighbors, etc, until the target is found or no neighbors remain.
Depth-First Search (DFS): an algorithm where you search for a path from a start node to a target by visiting a string of neighbors until you reach a dead end, then backtrack to the most recent unvisited neighbor; repeat until target is found or all neighbors have been visited
BFS and DFS runtimes: the runtimes depend on the number of edges in the graph. If we assume each node has constant # edges, both run in O(n).

Tractability

Brute force approach: an algorithmic strategy - solve a problem by generating all possible solutions and checking them.

Travelling Salesperson: a problem where you find the shortest route across all nodes in a graph. Runs in O(n!).
Puzzle Solving: a problem where you solve a jigsaw puzzle by finding an arrangement of pieces that fits all constraints. Runs in O(n!).
Subset Sum: a problem where you find a subset of numbers in a list that sums to a target number. Runs in $O(2^n)$.
Boolean Satisfiability: a problem where you find a combination of inputs that makes a circuit output 1. Runs in $O(2^n)$.
Exam Scheduling: a problem where you find an arrangement of exams across *k* timeslots such that no student has a conflict. Runs in $O(k^n)$.

Tractable: a problem is tractable if its worst-case runtime can be represented as a polynomial equation. Opposite is intractable.

Complexity class: a collection of function families that have similar efficiency for certain tasks and are bounded by (no worse than) a certain runtime.

P: a complexity class of problems that are tractable to *solve*
NP: a complexity class of problems that are tractable to *verify*
P vs NP: a big unsolved problem in CS. Are the complexity classes P and NP the same? We don't know!

Heuristic: a search technique used to find good-enough solutions to problems. Generates scores to choose next steps instead of using brute force.