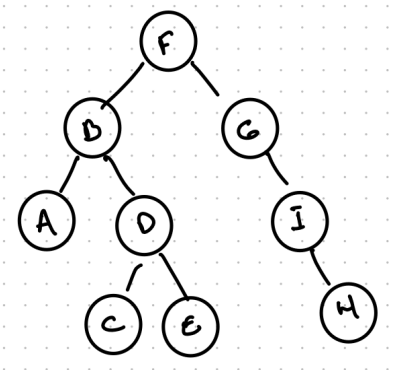
HW10: Graphs, Trees, Algorithms

HW instructions

Problem 1 Tree Vocab



The root of the Tree above is node F.

- A neighbor of node X is any node which shares an edge with X (e.g. G and I are neighbors while H and E are not).
- A node's parent is the unique neighbor which is closer to the root (the root has no parent). For example, B's parent is F.
- A sibling of node X is any node which shares a parent with X.
- An ancestor of node X is any node on the path from X to the root.
- X is a descendent of Y if Y is an ancestor of X.

Assume that the sibling, neighbor, ancestor and descendent relation are reflexive (e.g. every node is its own sibling) while the parent and child relations are not (no node is its own parent).

Express the set all the nodes which meet each of the following criteria.

i Child of D

ii Parent of I

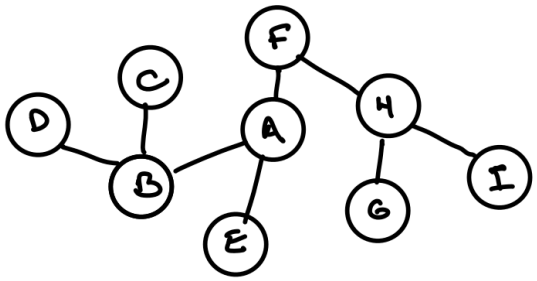
iii Sibling of D

iv Ancestor of H

v Descendent of B

vi the set of Neighbors of D or G which are not also neighbors of B

Problem 2 Graph Traversal 1



Give the order of nodes in each of the searches below, starting at the indicated node. When a method may select from among many next candidate nodes, prefer the node which is alphabetically first.

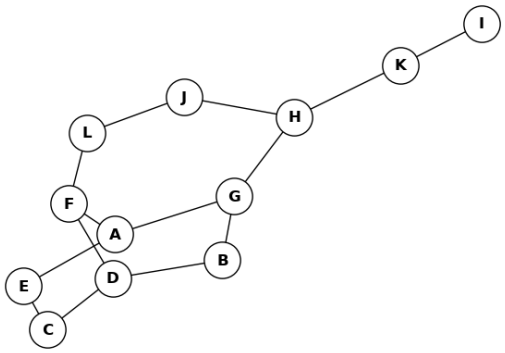
i Breadth First Search starting at node A

ii Depth First Search starting at node A

iii Breadth First Search starting at node F

iv Depth First Search starting at node F

Problem 3 Graph Traversal 2



Wherever possible below, select the node which is earlier in the alphabet first (e.g. prefer visiting node A first over node B, when the search allows you to visit either).

i Starting at I, find the Breadth-First-Search (BFS) ordering of nodes in the graph above.

ii Starting at B, find the Breadth-First-Search (BFS) ordering of nodes in the graph above.

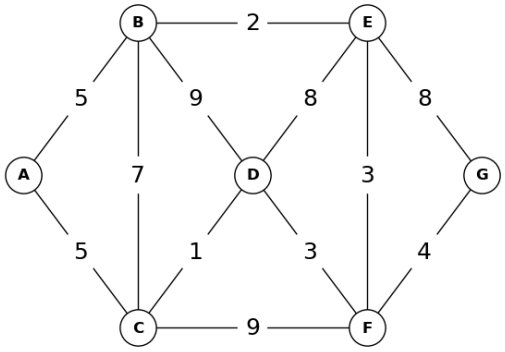
iii Starting at E, find the Breadth-First-Search (BFS) ordering of nodes in the graph above.

iv Starting at I, find the Depth-First-Search (DFS) ordering of nodes in the graph above.

v Starting at B, find the Depth-First-Search (DFS) ordering of nodes in the graph above.

vi Starting at E, find the Depth-First-Search (DFS) ordering of nodes in the graph above.

Problem 4 Dijkstra SP



Using Dijkstra's algorithm, find the shortest path from node A to G. Please provide a table which shows the path weight and predecessor from A to every node, labelling the visited node at each step. an example solution is given here.

Problem 5 Chain Characteristic

An undirected graph G is connected, has 8 vertices, and has degree of every vertex at most 2. Prove that there are only 2 such graphs, and draw them.

Problem 6 New Sorting

An array is *nearly-d sorted* if any element is not further than d spots from its sorted position. Consider the sorted list of elements:

$$X = [1, 5, 9, 10, 15, 20, 34, 57, 66, 91]$$

The same elements form a nearly-sorted list:

$$A = [1, 5, 10, 15, 9, 20, 34, 57, 91, 66]$$

with $d = 2$ because each value is, at most, 2 spots from its sorted position. Consider that the value 9 is out of order as it is in index 4[1] in A while it is in index 2 in X. Similarly,

$$B = [1, 5, 10, 9, 15, 20, 34, 57, 91, 66]$$

is nearly sorted with $d = 1$ as value 9 is in index 3 instead of index 2, 66 is in index 9 instead of index 8 and so on.

Bubble Sort[2], a sorting algorithm we have not covered, has an advantage over other methods when operating on nearly-d sorted lists.

i Describe Bubble Sort's Advantage in the best case scenario over other methods.

ii Bubble Sort need only pass through a nearly-d sorted list d times to ensure the list is sorted. Justify why this is the case. (Hint: consider the early termination condition of Bubble Sort)

Problem 7 Three Stooges

Moe, Larry and Curly have just purchased three new computers, each with its own processing speed and sorting algorithm:

[1] We adopt the Python convention of indexing $0, 1, 2, 3, \ldots$

[2] Wikipedia is a great place to start your Bubble Sort studying https://en.wikipedia.org/wiki/Bubble_sort, the animation in particular was instructive. However, more kinesthetic learners may appreciate the following video too: https://www.youtube.com/watch?v=lyZQPjUT5B4.

	Comparisons / sec	Search Algorithm	$T(n)$
Moe	50	Linear Search	n
Larry	5	Optimal Chunk Search	$2\sqrt{n}$
Curly	1	Binary Search	$\log_2 n$

where $T(n)$ is the number of comparisons it takes, in the worst case, to sort a list of size n.

Note that questions ii and iii below are not easily solved with pencil and paper, please show an initial equation and use Wolfram Alpha (https://www.wolframalpha.com/input?i=x%5E2+%2B+5x+%2B+6+%3D+17+x) to compute a final answer (the "solutions" box on the linked page may be helpful). There is a very similar example in recitation10, whose solutions you may access, which may also be helpful here.

i What is the smallest list input size n_0 (whole number) which ensures that for any $n \geq n_0$ and worst case list per method, Larry's computer sorts faster than Moe's?

ii What is the smallest list input size n_0 (whole number) which ensures that for any $n \geq n_0$ and worst case list per method, Curly's computer sorts faster than Moe's?

iii What is the smallest list input size n_0 (whole number) which ensures that for any $n \geq n_0$ and worst case list per method, Curly's computer sorts faster than Larry's?

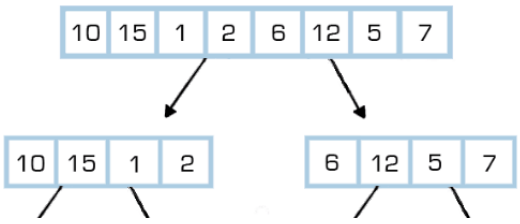
Problem 8 Sorting Steps

Sort the following list with the following algorithms, showing all the intermediate steps.

$$18, 45, 23, 2, -5, 10, 99, 0$$

i Insertion Sort
(Rewrite the list at each step, using the $\square$ symbol to divide the sorted portion of the array on its left from the unsorted portion to its right.)

ii Merge Sort
(Draw an array which shows the splitting and re-combining of the list. The first step, on another list of values, is shown below to demonstrate the intended notation)



Problem 9 Solve recurrence

Solve[3] the following recurrence equation:

$$T(n) = 9T(n-1) + 9 \quad \text{where } T(0) = 0$$

You may leave any series unsimplified in your solution (e.g. your answer can include terms like $\sum_{i=1}^{10} 2^i$)

[3] Express without recurrence