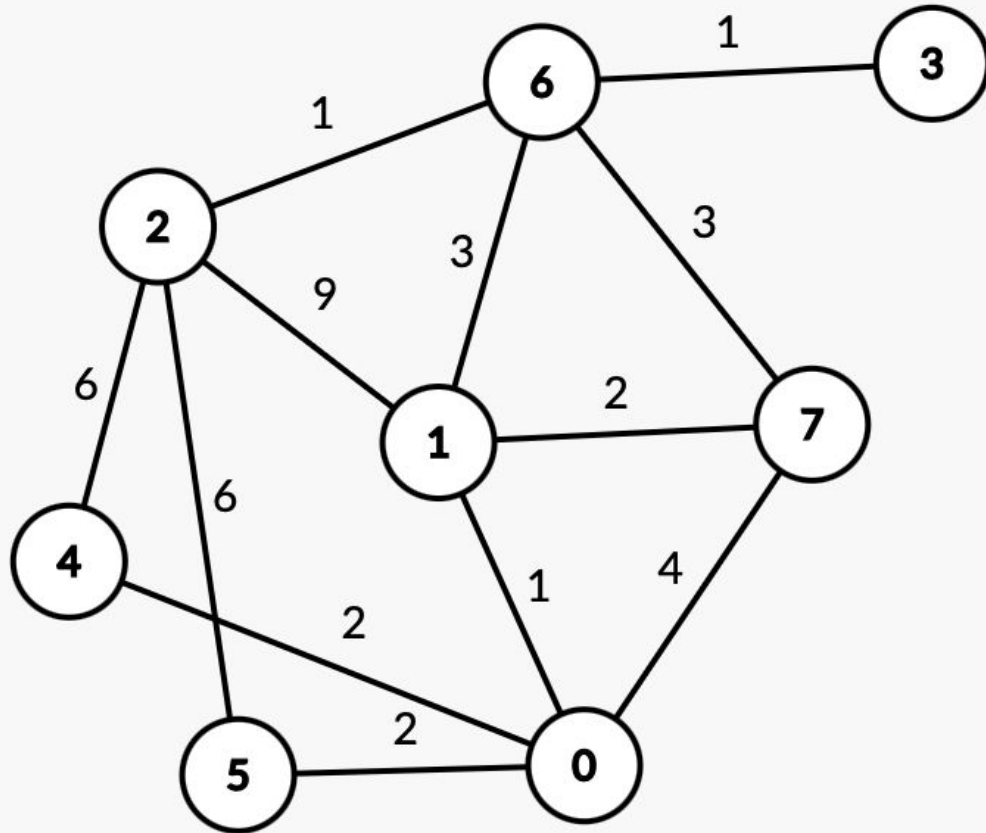


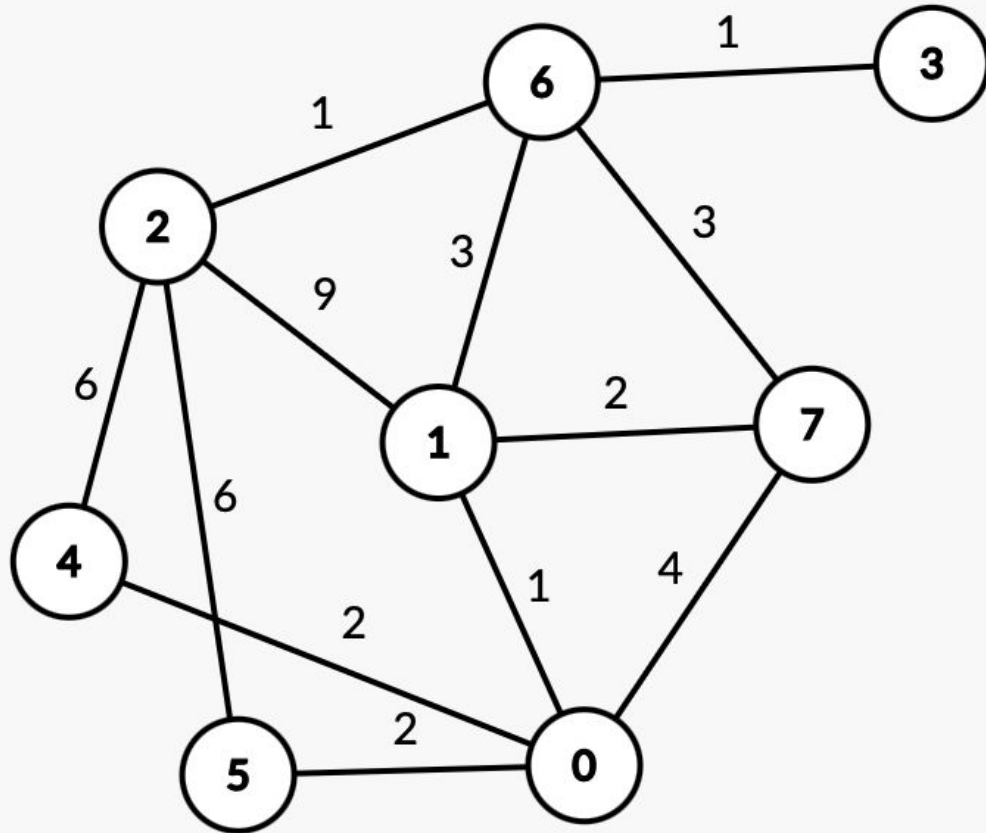
Final Exam Review

MST

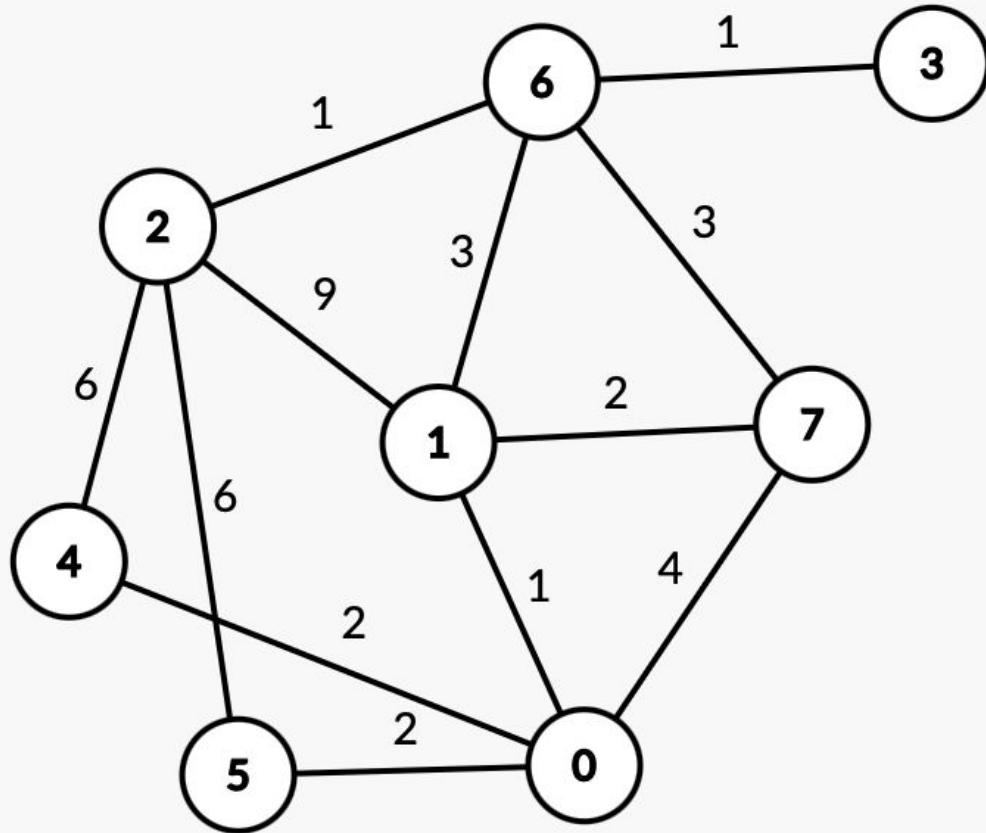
Run Baruvka's Algorithm on the graph below.



Run Kruskal's Algorithm on the graph below.

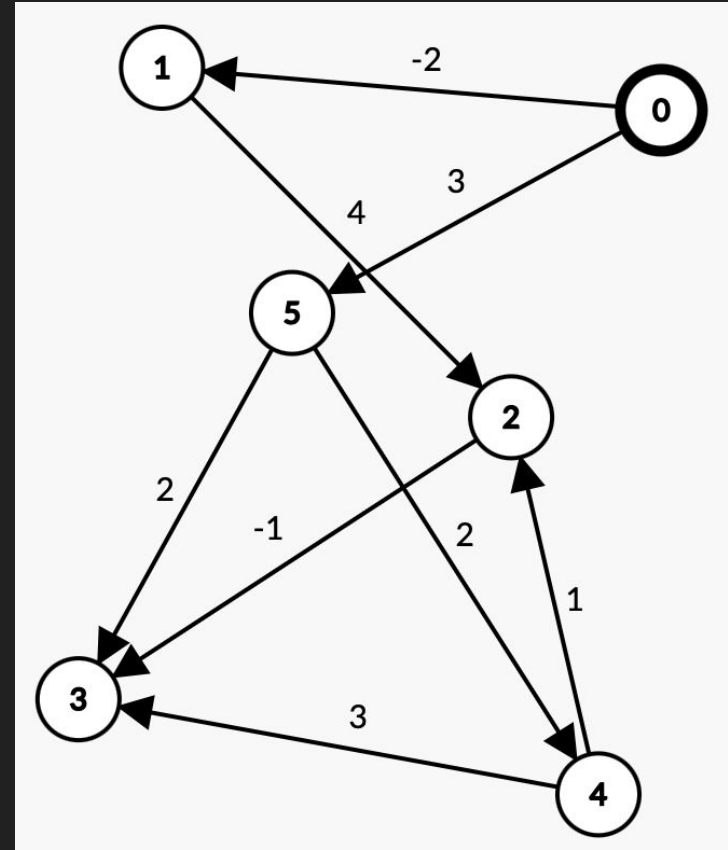


Run Prim's Algorithm on the graph below.

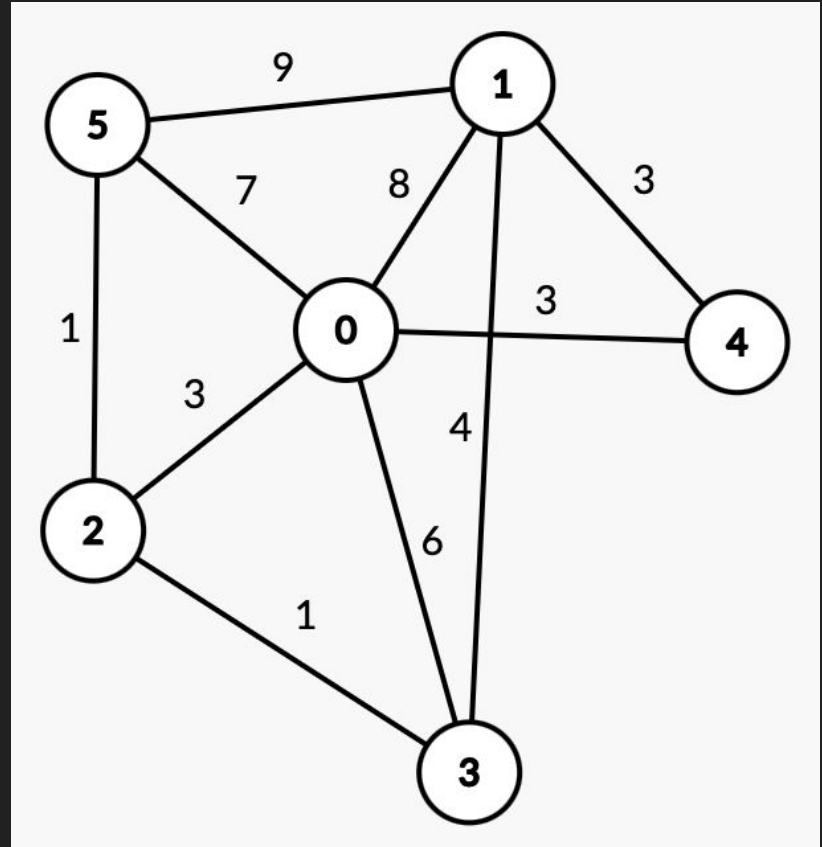


Shortest Path

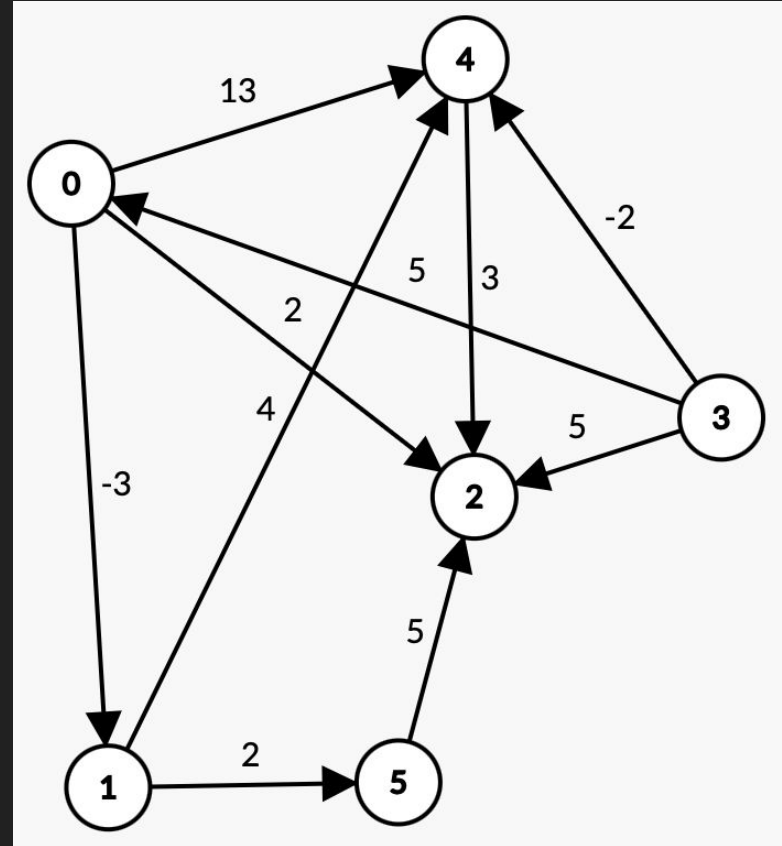
Determine which algorithm would be best for determining the shortest paths from vertex 0 to the other vertices in the graph below. Then use that algorithm to calculate the paths.



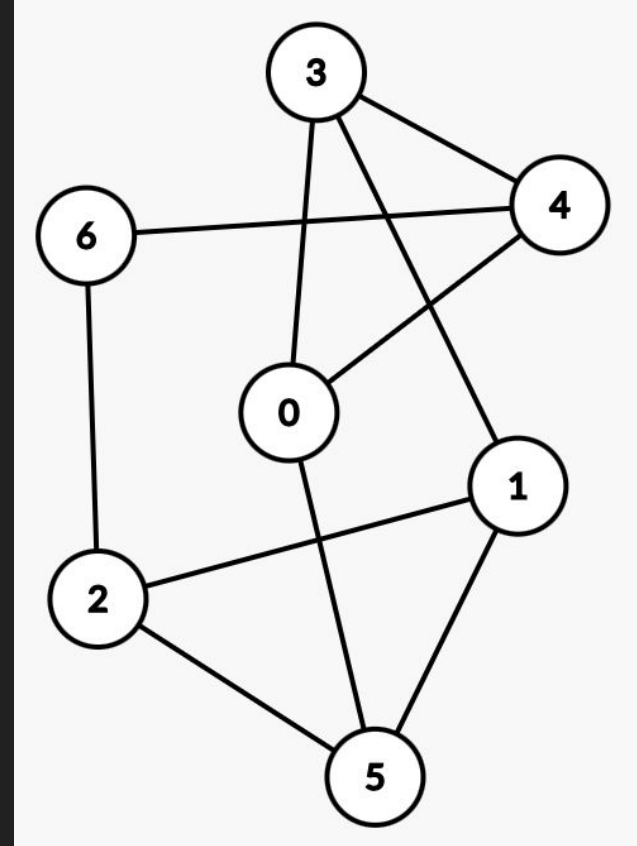
Determine which algorithm would be best for determining the shortest paths from vertex 0 to the other vertices in the graph below. Then use that algorithm to calculate the paths.



Determine which algorithm would be best for determining the shortest paths from vertex 0 to the other vertices in the graph below. Then use that algorithm to calculate the paths.



Determine which algorithm would be best for determining the shortest paths from vertex 0 to the other vertices in the graph below. Then use that algorithm to calculate the paths.



Union Find

Let U be a union find with 7 vertices. Show the contents of the id array and U as the following operations are executed. U uses Quick-Find.

union 0 6

union 1 2

union 0 3

union 4 6

union 5 1

union 3 5

Let U be a union find with 7 vertices. Show the contents of the id array and U as the following operations are executed. U uses Quick-Union.

union 0 6

union 1 2

union 0 3

union 4 6

union 5 1

union 3 5

Let U be a union find with 7 vertices. Show the contents of the id array and U as the following operations are executed. U uses Weighted Quick-Union.

union 0 6

union 1 2

union 0 3

union 4 6

union 5 1

union 3 5

Let U be a union find with 7 vertices. Show the contents of the id array and U as the following operations are executed. U uses Weighted Quick-Union with Path Compression.

union 0 6

union 1 2

union 0 3

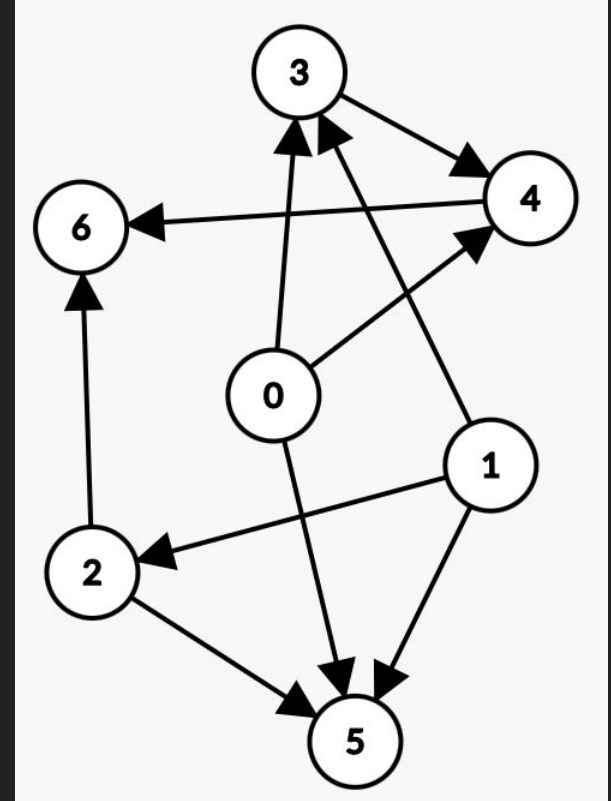
union 4 6

union 5 1

union 3 5

Directed Graphs

Determine if there are 0, 1, 2, or more than 2 topological orderings of this graph. If there is at least 1, list one.



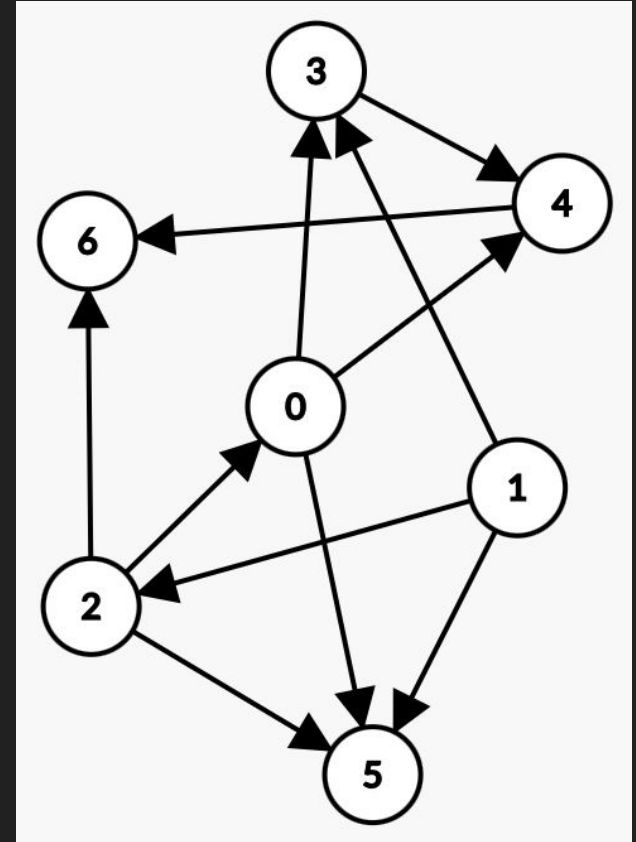
Determine if the following graph could be the transitive closure of another graph. If no, explain why not. If yes, write out everything you can conclude about the original graph.

0	0	1	1	1
0	0	1	1	0
0	0	1	0	0
0	1	1	0	0
1	1	1	0	0

Determine if the following graph could be the transitive closure of another graph. If no, explain why not. If yes, write out everything you can conclude about the original graph.

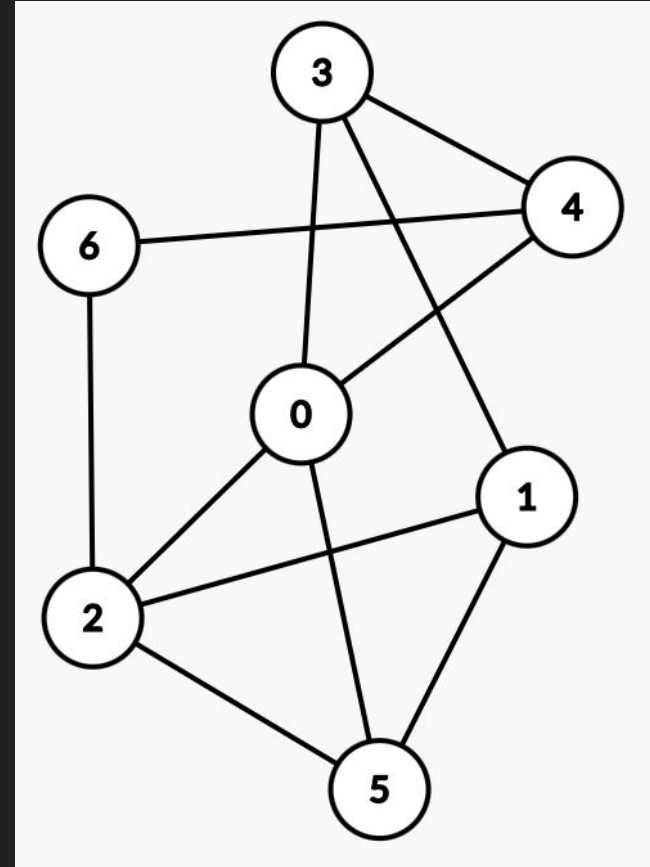
1	1	1	1	1
\emptyset	1	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	1	\emptyset	\emptyset
1	1	1	1	1
1	1	1	1	1

Is the following graph strongly connected? If not, are there any strongly connected components?

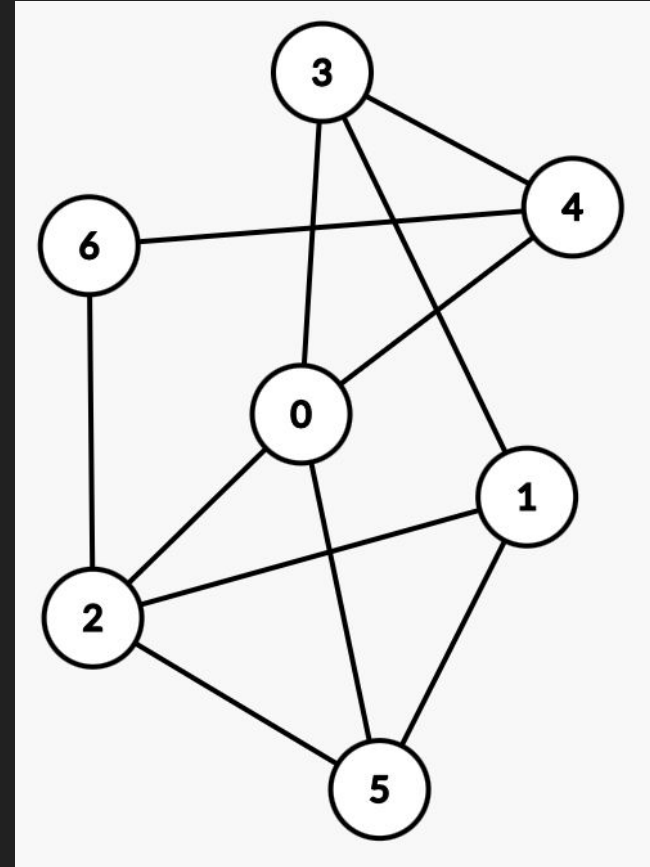


Graph Traversals

Run DFS on the following graph starting at vertex 0 and determine the order in which the vertices are discovered. Assume that neighbors are examined in numerical order.



Run BFS on the following graph starting at vertex 0 and determine the order in which the vertices are discovered. Assume that neighbors are examined in numerical order.



Searching

Build a Skiplist by inserting the following items. The “random” level is given in parentheses. Also, tell what the probability is of inserting N level-0 nodes in a row.

insert 6 (0), insert 7 (0), insert 1 (2), insert 2 (1), insert 3 (0), insert 9 (1), insert 10 (2), insert 8 (0)

Build an LLRB Tree by executing the following operations. Also, tell what the maximum height of any LLRB Tree with N nodes is and explain your reasoning..

insert 6, insert 0, insert 2, insert 1, delete 2, insert 8, insert 4, insert 5, delete 8

Build an 2-3 Tree by executing the following operations. Also, tell what the maximum and minimum heights of any 2-3 Tree with N nodes is and explain your reasoning.

insert 6, insert 0, insert 2, insert 1, delete 2, insert 8, insert 4, insert 5, delete 8

Build a BST by executing the following operations. Also, tell what the maximum and minimum heights of any BST with N nodes is and explain your reasoning.

insert 6, insert 0, insert 2, insert 1, delete 2, insert 8, insert 4, insert 5, delete 8

Explain the difference between separate chaining, linear probing, and quadratic probing. Be sure to include a discussion of the runtime of operations and space usage.

Compare and contrast all the search structures we discussed: Hashtables, BSTs, 2-3 Trees, LLRB Trees, and Skiplists.

Which is better: Using a binary search tree or using a sorted array and using binary search? Explain your reasoning.

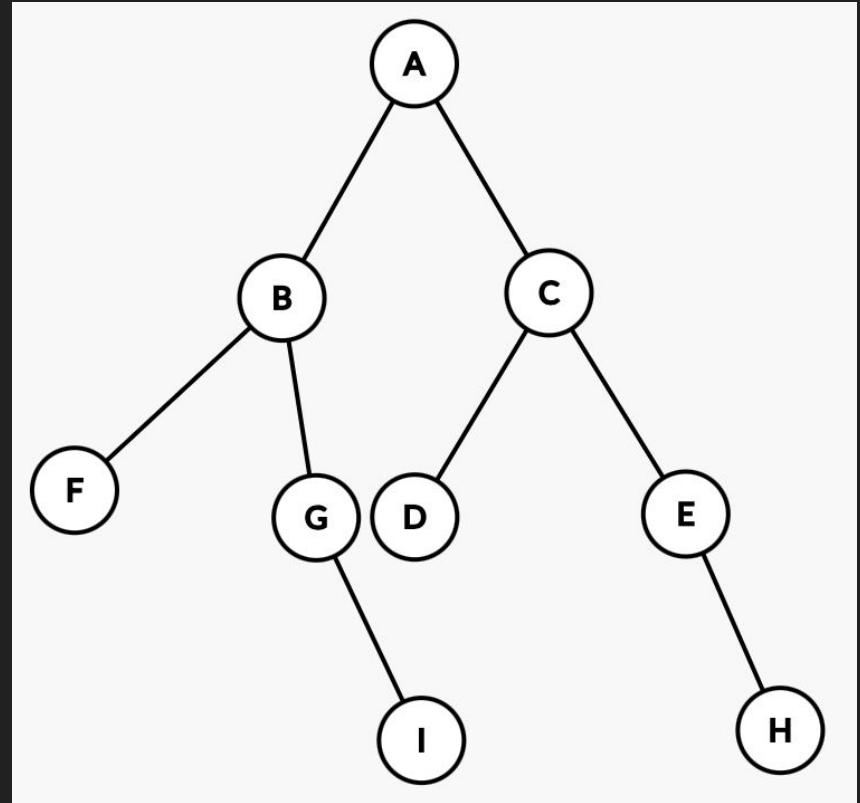
Sorting

Explain the best and worst-case runtimes of each of the sorting algorithms we discussed in class: Bubble sort, Selection Sort, Insertion sort, Heapsort, Shellsort, Mergesort, Quicksort, Bucket Sort, Radix Sort

Sort the following array using: Bubble sort, Selection Sort, Insertion Sort, Shellsort, Heapsort, Mergesort, Quicksort.

8	1	0	2	6	7
---	---	---	---	---	---

Give the pre-order, post-order, in-order, bfs, and dfs traversals of the tree below.



Why is Quicksort considered “quick” when it’s worst-case runtime is quadratic?

Basic Data Structures

Build a min binary heap with the following commands and show it as an array.

put 8, put 1, put 2, put 3, delMin, put 7, put 0, delMin

Compare and contrast a linked-list-based Stack/Queue with an array-based Stack/Queue.

Explain how the amortized cost of adding to the end of an ArrayList in Java can be $O(1)$ when it is built with an array and has to be resized when full.

Explain how a Queue can be implemented with an array without wasting too much time or space.

Other Things You Should Review

- Structural Induction
- Analyzing Algorithms
- Recurrence Relations
- Proofs: especially inductive proofs
- big-Oh, big-Omega, big-Theta