Checkpoint 3 study guide

Searching & Graphs

Information

- Checkpoint 3 is Weds, Dec 3 in class (our last day of class).
- As usual, you can bring one hand-written (ok hand-written on tablets and then printed) back and front sheet of paper (i.e. two pages). NO slides shrunk and copy pasted.
- Review lecture slides along with slides on practice problems and links to code. Go over quizzes, labs, and assignments. Use the practice problems in this presentation. If you want to read in more depth, use the recommended textbook (extra copies for in-lab use in the dept library)
- Practice writing code on paper.

Review

Checkpoint III Review

- LLRB Trees
- Hashing
- Graphs
  - BFS/DFS
  - Shortest path
  - MSTs
  - DAGs
- Choosing data structures
- Practice problems
- Answers

LLRBs

- Definitions, Search, Insertion for LLRBs
  - Equivalence with 2-3 trees
- Performance

Hash tables

- Chapter 3.3 (Pages 458-477).
- Hashing, separate chaining, open addressing.

Summary for Dictionaries

- Worst case search and insert are $O(n)$ for BSTs. Not great!

	Ordered Operations	Worst case			Average case		
		Search	Insert	Delete	Search	Insert	Delete
BST	Yes	n	n	n	$\log n$	$\log n$	$\log n$
balanced search trees	Yes	$\log n$	$\log n$	$\log n$	$\log n$	$\log n$	$\log n$
hash tables with separate chaining	No	n	n	n	1	1	1
hash tables with linear probing	No	n	n	n	1	1	1

Undirected Graphs

- Chapter 4.1 (Pages 515-556).
- Definitions, representations, APIs.
- BFS.
- DFS.
- Textbook code.
  - https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Graph.java.html
  - https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DepthFirstSearch.java.html
  - https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/BreadthFirstPaths.java.html

Directed Graphs

- Chapter 4.2 (Pages 566-594).
- Definitions, representations, APIs.
- BFS.
- DFS.
- Textbook code.
  - https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Digraph.java.html
  - https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DirectedDFS.java.html

Shortest Paths

- Chapter 4.4 (Pages 638-676).
- Dijkstra’s algorithm.
- https://visualgo.net/en/sssp

Minimum Spanning Trees

- Chapter 4.3 (Pages 604-629).
- Know how to apply Kruskal's and Prim's algorithms.
  - https://algs4.cs.princeton.edu/43mst/
  - https://visualgo.net/en/mst

Problem	**Problem Description**	**Solution**	**Efficiency**
paths	Find a path from s to every reachable vertex.	DFS	O(V+E) time Θ(V) space
shortest paths	Find the shortest path from s to every reachable vertex.	BFS	O(V+E) time Θ(V) space
shortest weighted paths	Find the shortest path, considering weights, from s to every reachable vertex.	Dijkstra's	O(E log V) time Θ(V) space
minimum spanning tree	Find the minimum spanning tree.	Prim's	O(E log V) time Θ(V) space
minimum spanning tree	Find the minimum spanning tree.	Kruskal's	O(E log E) time Θ(E) space

Directed Acyclic Graphs

- What is a DAG, topological sorting, shortest paths with negative edge weights

Problem	Problem Description	Solution	Efficiency
topological sort	Find an ordering of vertices that respects edges of our DAG.	DFS from source nodes	O(V+E) time Θ(V) space
DAG shortest paths	Find a SPT on a DAG. Negative weights OK.	topological sort, then visit vertices in that order and relax	O(V+E) time Θ(V) space
longest paths	Find a longest paths tree on a graph.	No good known solution exists.	?
DAG longest paths	Find a longest paths tree on a DAG.	flip signs, then DAG shortest paths, flip signs again	O(V+E) time Θ(V) space

Choosing data structures

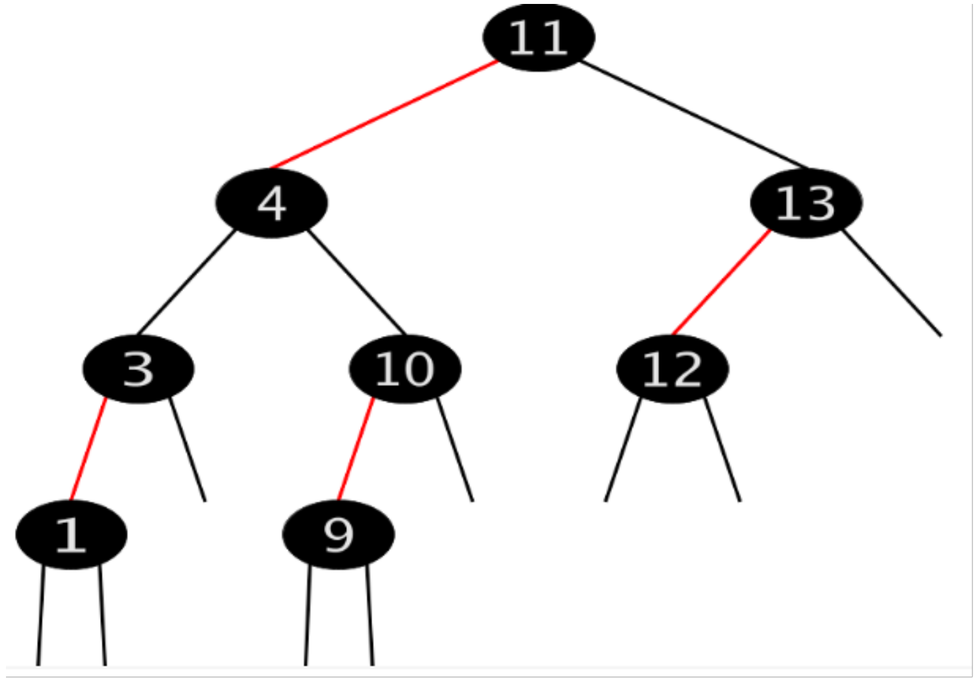
- Given a word problem, what data structure (out of all the ones we've seen) would you choose to solve it? (Similar to our coding interview practice lab and your final project)
  - Linear data structures
  - Stacks
  - Heaps
  - Queue vs priority queue
  - Binary trees
  - Binary search trees
  - Hashtable
  - Graphs (undirected/directed)
  - etc...

Practice problems

Problem 1: LLRBs

You are given the following LLRB.

Draw the final LLRB tree and the intermediate trees and name the different elementary operations you performed (insert, color flip, rotate left, rotate right) after you insert the key **8** in the provided LLRB.



Problem 2: Hashtables

Consider inserting the keys 25, 17, 12, 26, 27, 5, 9, 29, 11, 23 into a hash table of size $m = 11$ using the hash function $h(k) = k \,\%\, m$. For each of the following questions, fill in the following hash table:

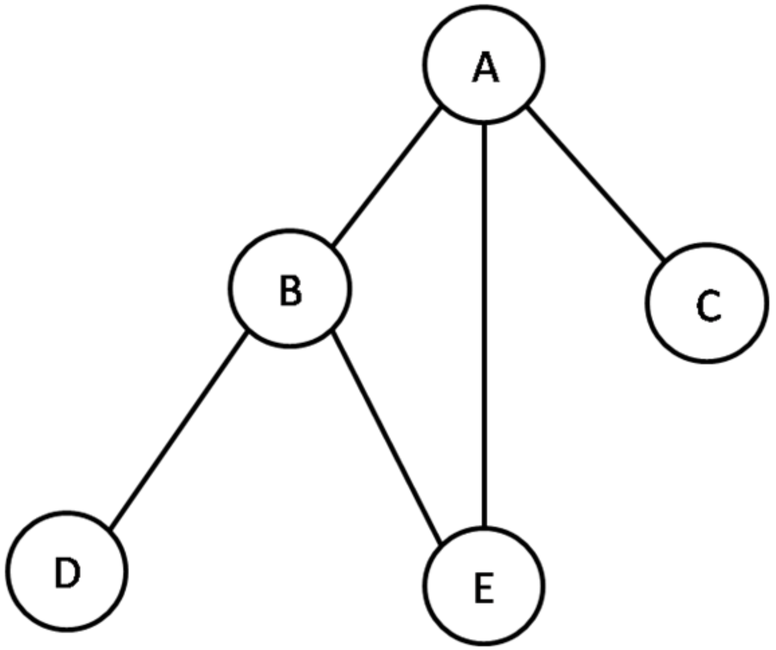
a. using open addressing and linear probing with $h(k, i) = (h(k) + i) \,\%\, m$.

b. using open addressing and quadratic probing with $h(k, i) = (h(k) + i^2) \,\%\, m$.

c. using separate chaining (insert in the front of the chain).

0	1	2	3	4	5	6	7	8	9	10

Problem 3: Traversing Graphs



Show the adjacency matrix and adjacency list representations of the undirected graph above.

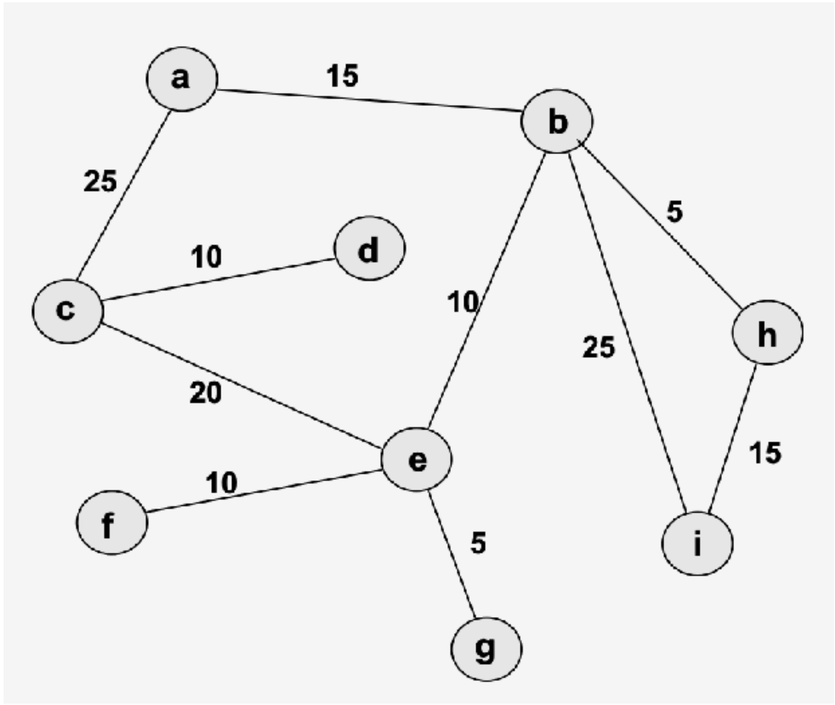
Run a. DFS and b. BFS starting at vertex A assuming adjacent vertices are returned in lexicographic order. Fill in the table below:

v	marked	distTo	edgeTo
A			
B			
C			
D			
E			

Problem 4: Shortest Paths

Run Dijkstra's algorithm on this graph, starting at vertex a. Fill in the resulting distTo[] and edgeTo[] arrays below. In the edgeTo[] column, please indicate the last edge in the shortest path from a to every other vertex and mark the shortest path tree.

v	distTo	edgeTo
a		
b		
c		
d		
e		
f		
g		
h		
i		



Problem 5: Data structures

For each of the following problems, please choose the most appropriate data structure to use.

Many text editors provide the possibility for the user to undo an arbitrary number of commands. What data structure should the implementers use to save past commands in order to provide this capability?

Stack Queue Tree Priority Queue

Some word processors collect embedded footnotes in text and print them all (in the order they appeared) at the end of the document. Which data structure should the implementers of the word-processor use to save the footnotes as they are encountered?

Stack Queue Tree Priority Queue

Which data structure does your computer use to save the directory structure of its file system?

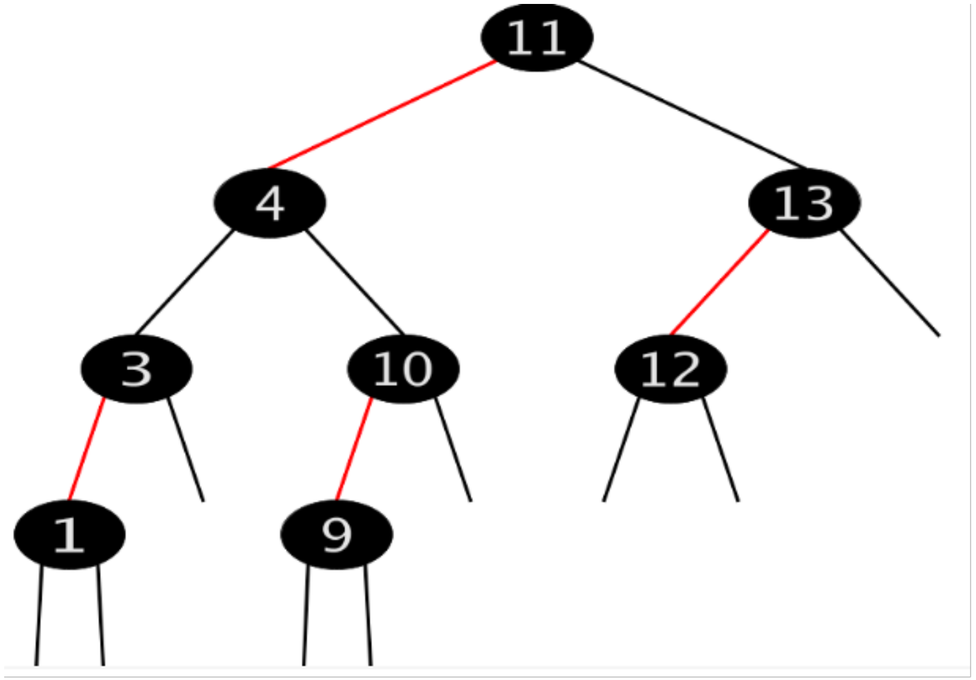
Stack Queue Tree Priority Queue

Solutions

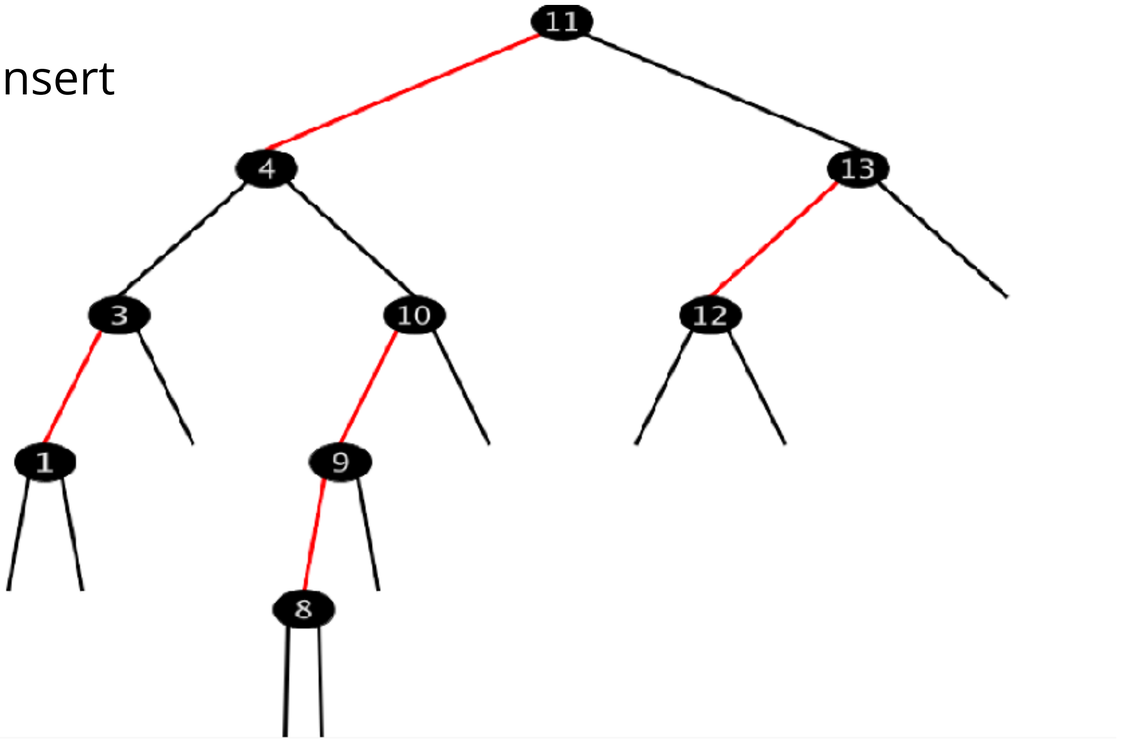
Solution to Problem 1: LLRBs

You are given the following LLRB.

Draw the final LLRB tree and the intermediate trees and name the different elementary operations you performed (insert, color flip, rotate left, rotate right) after you insert the key **8** in the provided LLRB.



1. Insert

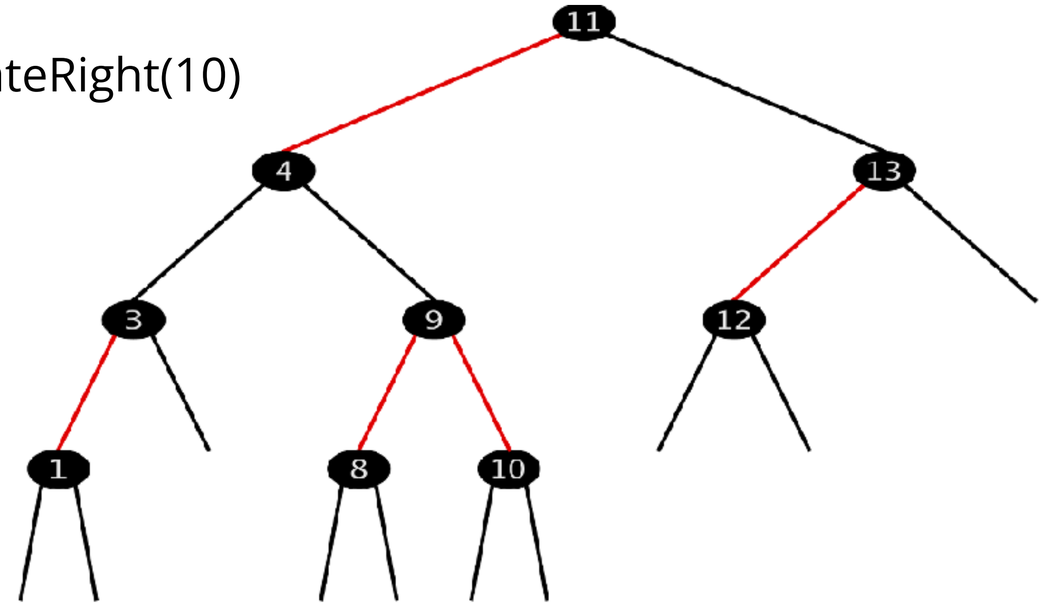


Solution to Problem 1: LLRBs

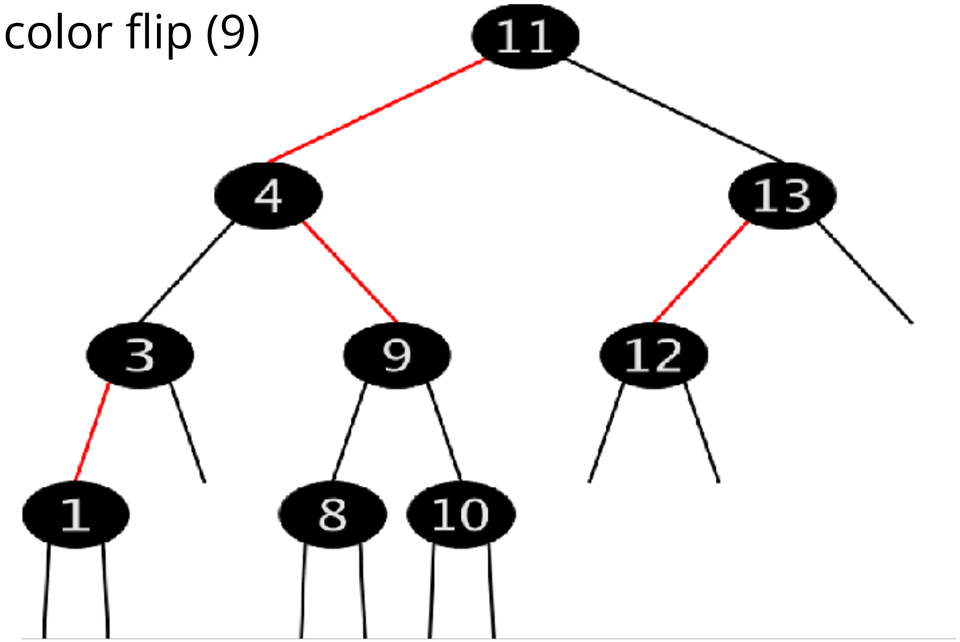
You are given the following LLRB.

Draw the final LLRB tree and the intermediate trees and name the different elementary operations you performed (color flip, rotate left, rotate right) after you insert the key **8** in the provided LLRB.

2. rotateRight(10)



3. color flip (9)

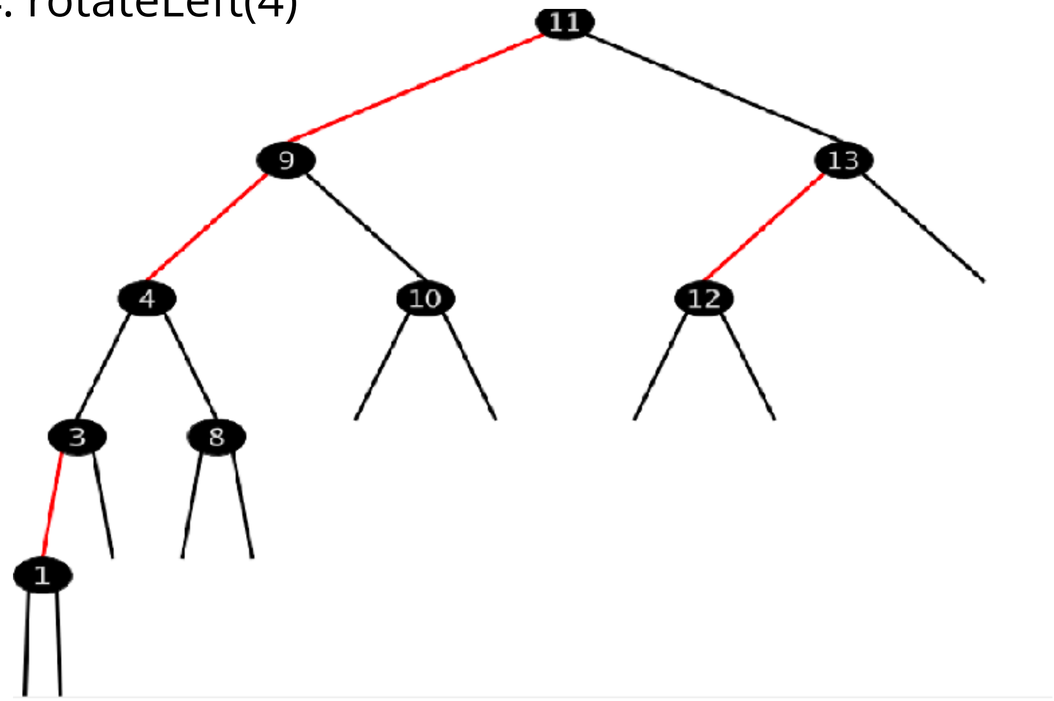


Solution to Problem 1: LLRBs

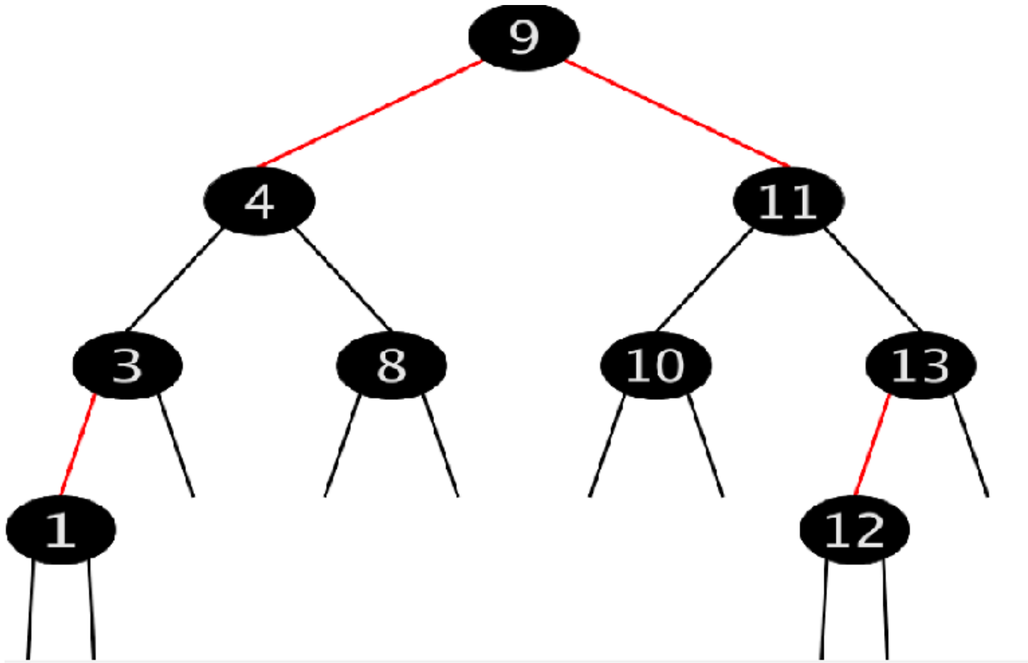
You are given the following LLRB.

Draw the final LLRB tree and the intermediate trees and name the different elementary operations you performed (color flip, rotate left, rotate right) after you insert the key **8** in the provided LLRB.

4. rotateLeft(4)



5. rotateRight(11)

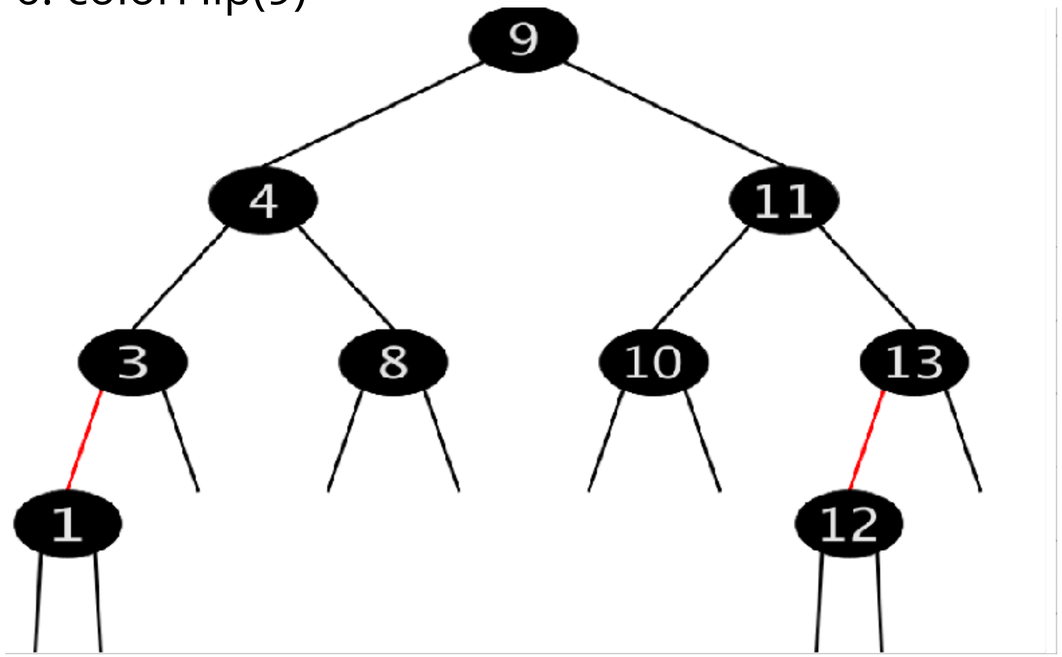


Solution to Problem 1: LLRBs

You are given the following LLRB.

Draw the final LLRB tree and the intermediate trees and name the different elementary operations you performed (color flip, rotate left, rotate right) after you insert the key **8** in the provided LLRB.

6. colorFlip(9)



Done!

Solution to Problem 2a: Hashtables

- Consider inserting the keys 25, 17, 12, 26, 27, 5, 9, 29, 11, 23 into a hash table of size $m = 11$ using the hash function $h(k) = k \,\%\, m$. For each of the following questions, fill in the following hash table using open addressing and linear probing with $h(k, i) = (h(k) + i) \,\%\, m$.

11	12	23	25	26	27	17	5	29	9	
0	1	2	3	4	5	6	7	8	9	10

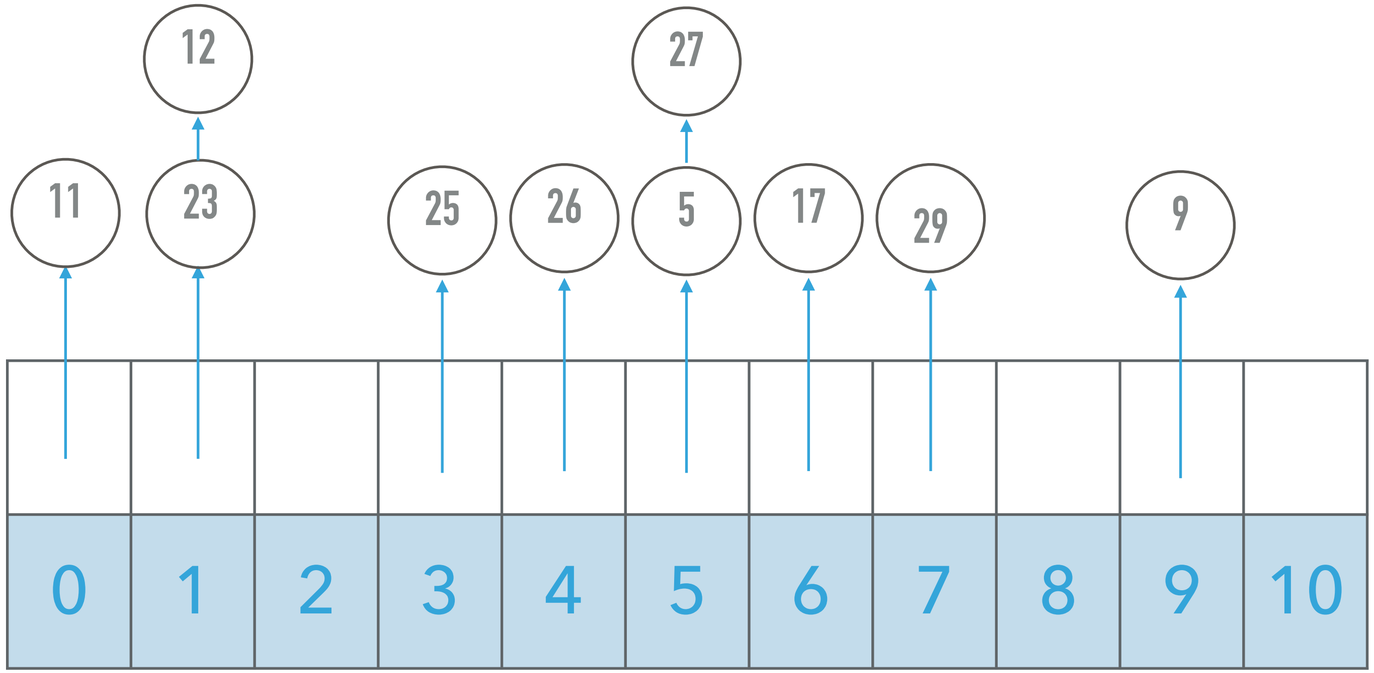
Solution to Problem 2b: Hashtables

- Consider inserting the keys 25, 17, 12, 26, 27, 5, 9, 29, 11, 23 into a hash table of size $m = 11$ using the hash function $h(k) = k \,\%\, m$. For each of the following questions, fill in the following hash table using open addressing and quadratic probing with $h(k, i) = (h(k) + i^2) \,\%\, m$.

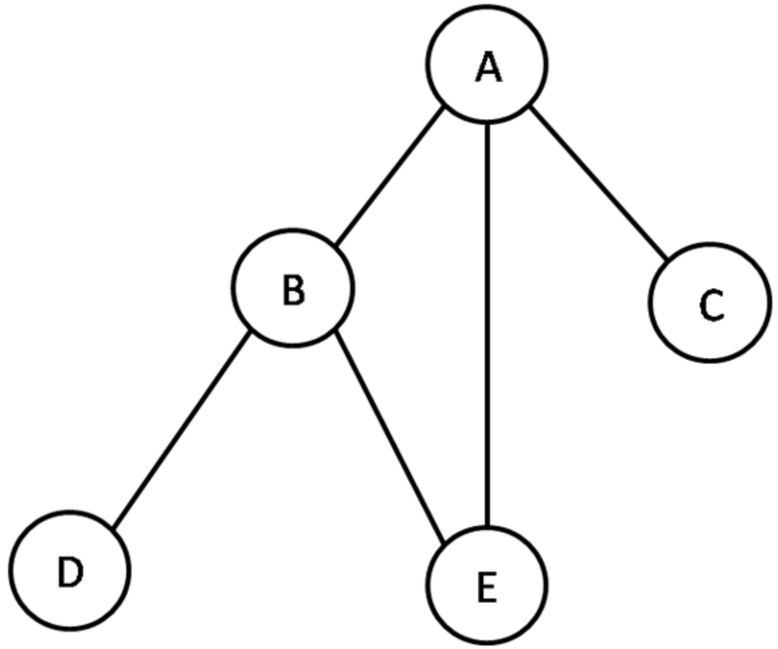
11	12	23	25	26	27	17	29		5	9
0	1	2	3	4	5	6	7	8	9	10

Solution to Problem 2c: Hashtables

- Consider inserting the keys 25, 17, 12, 26, 27, 5, 9, 29, 11, 23 into a hash table of size $m = 11$ using the hash function $h(k) = k \,\%\, m$. For each of the following questions, fill in the following hash table using separate chaining.



Solution to Problem 3a: Traversing Graphs



- Show the adjacency matrix and adjacency list representations of the undirected graph above.
- Run a. DFS assuming adjacent vertices are returned in lexicographic order.
- Order of visit: A, B, D, E, C

adj matrix

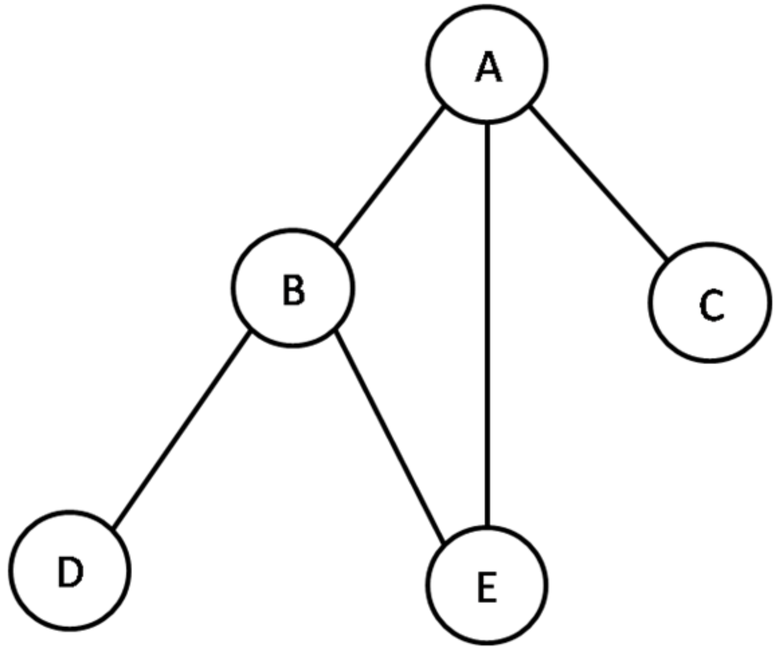
	A	B	C	D	E
A	0	1	1	0	1
B	1	0	0	1	1
C	1	0	0	0	0
D	0	1	0	0	0
E	1	1	0	0	0

adj list

A -> B, E, C
B -> A, D, E
C -> A
D -> B
E -> A, B

v	marked	edgeTo
A	**T**	**-**
B	T	A
C	T	A
D	T	B
E	T	B

Solution to Problem 3b: Traversing Graphs



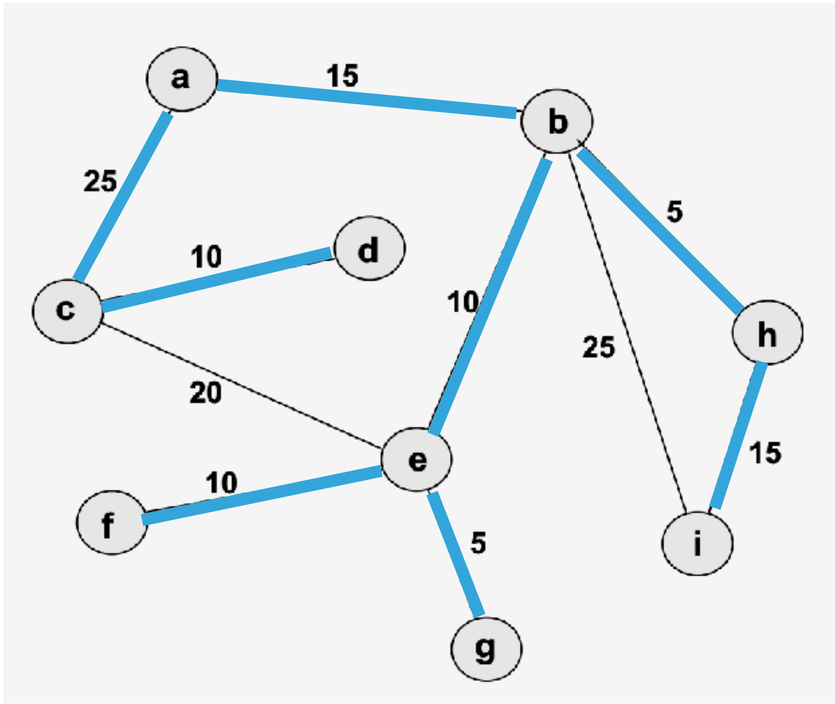
- Show the adjacency matrix and adjacency list representations of the undirected graph above.
- Run b. BFS assuming adjacent vertices are returned in lexicographic order.
- Order of visit: A, B, C, E, D

v	marked	distTo	edgeTo
A	**T**	**0**	**-**
B	T	1	A
C	T	1	A
D	T	2	B
E	T	1	A

Solution to Problem 4: Shortest Paths

Run Dijkstra's algorithm on this graph, starting at vertex a. Fill in the resulting distTo[] and edgeTo[] arrays below. In the edgeTo[] column, please indicate the last edge in the shortest path from a to every other vertex and mark the shortest path tree.

v	distTo	edgeTo
a	**0**	**-**
b	15	a
c	25	a
d	35	c
e	25	b
f	35	e
g	30	e
h	20	b
i	35	h



Solution to Problem 5: Data structures

For each of the following problems, please choose the most appropriate data structure to use.

Many text editors provide the possibility for the user to undo an arbitrary number of commands. What data structure should the implementers use to save past commands in order to provide this capability?

Stack Queue Tree Priority Queue

Some word processors collect embedded footnotes in text and print them all (in the order they appeared) at the end of the document. Which data structure should the implementers of the word-processor use to save the footnotes as they are encountered?

Stack **Queue** Tree Priority Queue

Which data structure does your computer use to save the directory structure of its file system?

Stack Queue **Tree** Priority Queue