Written Exam 2 (Sample)

Name:

Instructions

1. Write your name at the top of the *first* page and your initials at the bottom of *every* page.
2. Do *not* staple the exam.
3. Return the exam with *all* the pages, arranged in *ascending* order.
4. This is a closed-book exam. No form of communication is permitted (eg, talking, texting, etc.) during the exam, except with the course staff.
5. No electronic devices are permitted.
6. There are 25 multiple-choice questions in this exam, each worth 3 points.
7. Each question has *exactly one* correct answer, which must be *clearly* marked in the circle provided. Your answer will be considered incorrect otherwise.
8. You may use the blank spaces for any scratch work.
9. Discussing the exam contents with anyone who has not taken the exam is a violation of the academic honesty code.

Problem 1. Consider inserting the following key-value pairs in that order into a symbol table `st`.

`key:`	R	Q	J	G	L	R	M	I	Q	H	R	V
`value:`	1	2	3	4	5	6	7	8	9	10	11	12

a. What is the value returned by `st.size()`?

R Q J G L M I H V —> 9 size

~~1~~ ~~2~~ 3 4 5 7 8 10 12

~~6~~ 9

11

Ⓐ 12

Ⓑ 11

[✓ Ⓒ 9]

Ⓓ 8

Ⓔ 10

b. What is the value returned by `st.get("R")`?

R —> 11

b/c R —> 6 gets overwritten

Ⓐ 6

[✓ Ⓑ 11]

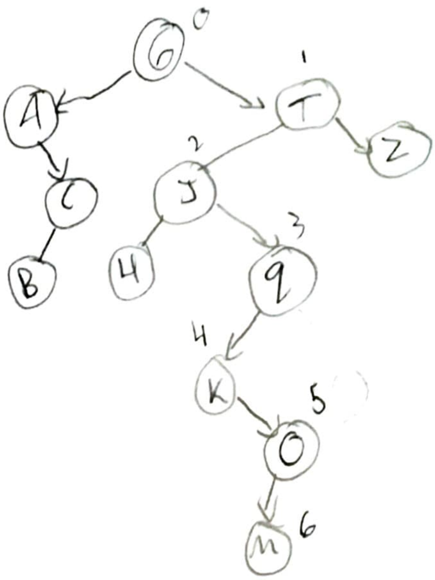
Ⓒ 3

Ⓓ 18

Ⓔ 1

Problem 2. Consider inserting the following keys (assume values to be non `null` and arbitrary) into a binary search tree (BST) symbol table `st`, an object of type `BST`.

G T J Q H Z K A O C M B



a. What is the height of the BST (assume root to be at height 0)?

(A) 5

(B) 7

[✓ (C) 6]

(D) 4

(E) 8

b. What is the value returned by `st.rank("M")`?

[✓ (A) 7]

(B) 5

(C) 8

(D) 6

(E) 4

Number of Nodes Smaller

K, J, H, G, A, C, B

c. What is the order in which the keys are visited if we traverse the BST in pre-order?

(A) A B C G H J K M O Q T Z

(B) G A C B T J H Q O K Z M

[✓ (C) G A C B T J H Q K O M Z]

(D) B C A H M O K Q J Z T G

(E) G A C B T J H M K Z O Q

1. Visit Root
2. Go left (recursive)
3. Go Right (recursive)

d. What is the order in which the keys are visited if we traverse the BST in in-order?

[✓ (A) A B C G H J K M O Q T Z]

(B) A B C G H J K M Z Q O T

(C) A B C G H J K Q Z T O M

(D) B C A H M O K Q J Z T G

(E) G A C B T J H Q K O M Z

1. Go left
2. Visit Root
3. Go right

e. What is the order in which the keys are visited if we traverse the BST in post-order?

(A) B C A H M O K Z Q J G T

(B) B C A H M O K J G Q Z T

(C) A B C G H J K M O Q T Z

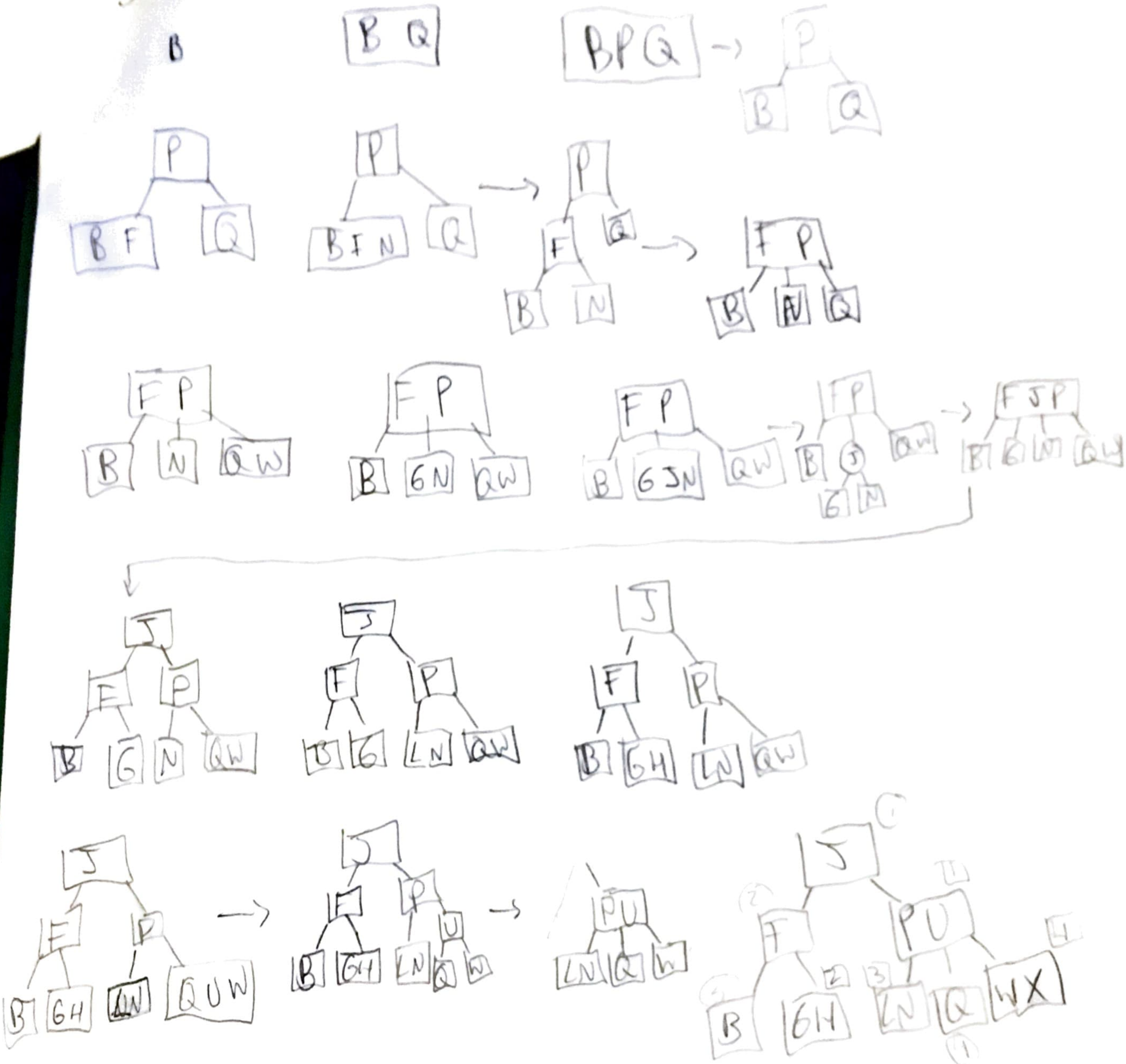
[✓ (D) B C A H M O K Q J Z T G]

(E) G A C B T J H Q K O M Z

1. Go left
2. Go right
3. Visit Node

Initials:

2-3 Search Tree



Problem 3. Consider inserting the following keys into an initially empty 2-3 search tree.

`B Q P F N W G J L H U X`

a. What is the height of the tree that results (assume root to be at height zero)?

(A) 3

(B) 5

(C) 4

(D) 1

[✓ (E) 2]

b. How many nodes does the tree contain?

(A) 6

[✓ (B) 8]

(C) 5

(D) 9

(E) 7

c. How many 2-nodes does the tree contain?

[✓ (A) 4]

(B) 6

(C) 5

(D) 3

(E) 7

d. How many 3-nodes does the tree contain?

(A) 6

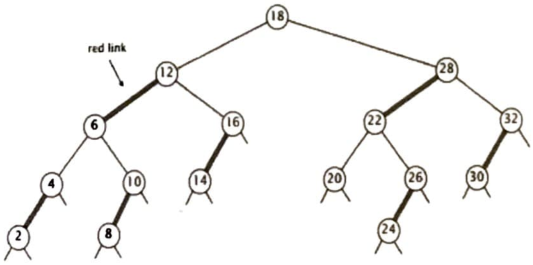
(B) 5

[✓ (C) 4]

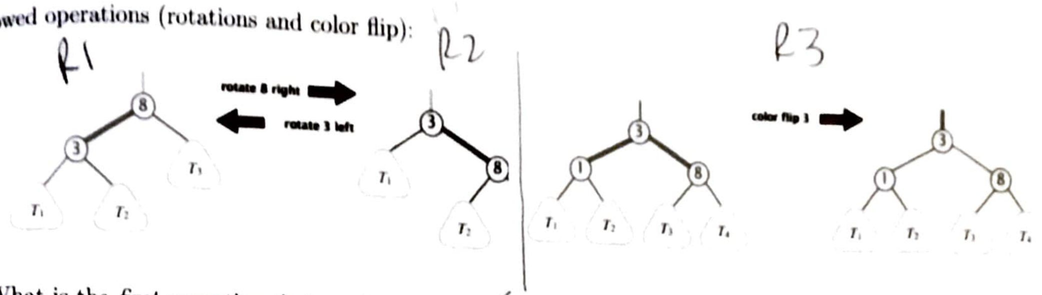
(D) 3

(E) 7

Problem 4. Suppose you insert the key 9 into the following left-leaning red-black BST:



Allowed operations (rotations and color flip):



a. What is the *first* operation that results?

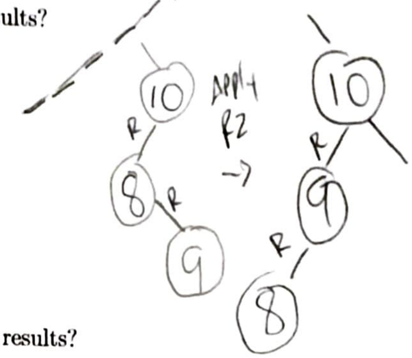
[✓ (A) Rotate 8 left]

(B) Rotate 10 right

(C) Rotate 12 right

(D) Rotate 6 left

(E) Color flip 9



b. What is the *second* operation that results?

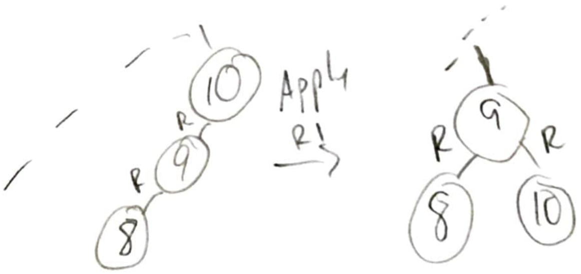
(A) Rotate 8 left

[✓ (B) Rotate 10 right]

(C) Rotate 12 right

(D) Rotate 6 left

(E) Color flip 9



c. What is the *third* operation that results?

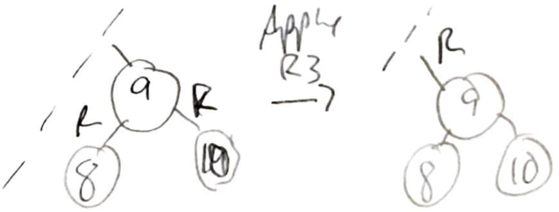
(A) Rotate 8 left

(B) Rotate 12 right

(C) Rotate 10 right

(D) Rotate 6 left

[✓ (E) Color flip 9]



d. What is the *fourth* operation that results?

(A) Rotate 8 left

[✓ (B) Rotate 6 left]

(C) Rotate 12 right

(D) Rotate 10 right

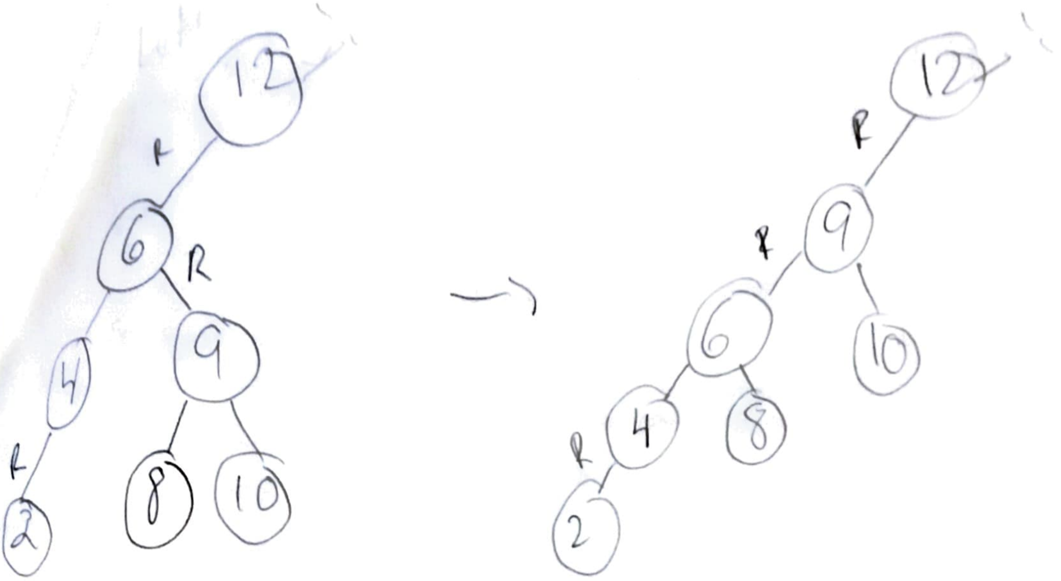
(E) Color flip 9

on next page

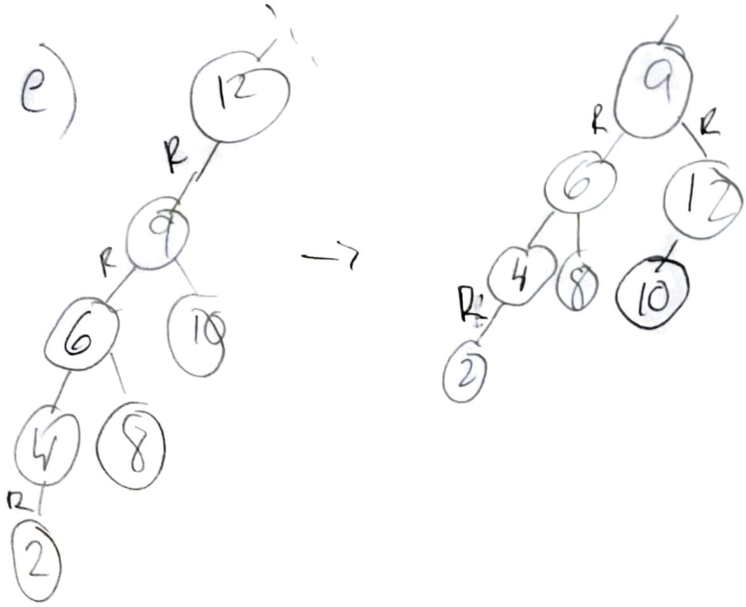
e. What is the *fifth* operation that results?

[✓ (A) Rotate 12 right]

Initials:



e)



(B) `Rotate 6 left`

(C) `Rotate 10 right`

(D) `Color flip 9`

(E) `Rotate 8 left`

Problem 5. Consider inserting the following keys (assume values to be non `null` and arbitrary) into an initially empty hash table of $M = 5$ lists, using separate chaining. Use the hash function $h(k) = k \mod M$ to transform the kth letter of the alphabet into a table index, where $1 \leq k \leq 26$.

10	4	23	5	22	21	12	16	6	11	24	25
J	D	W	E	V	U	L	P	F	K	X	Y

% 5

a. What is the length of the longest chain?

(A) 1

(B) 3

(C) 5

[✓ (D)] 4

(E) 2

0	JEY
1	UPFK
2	VL
3	W
4	DX

b. Which of the following keys is in the longest chain?

[✓ (A)] U

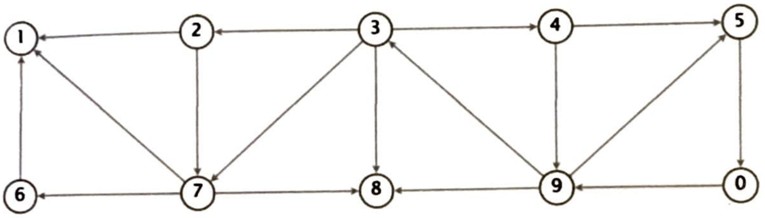
(B) O

(C) V

(D) J

(E) W

Problem 6. Perform a depth-first search in the digraph below, starting from vertex 0. Assume the adjacency lists are in sorted order: for example, when iterating over the edges pointing from 3, process the edge 3 → 2 before either 3 → 7 or 3 → 8.



a. List all vertices in pre-order.

(A) 0 9 3 7 8 4 1 2 6 5

(B) 0 9 3 6 1 5 4 7 8 2

[✓ (C)] 0 9 3 2 1 7 6 8 4 5

~~(D) 0 9 3 8 2 6 4 5 1 7~~

- Go as far as you can in natural order
- if no options, go back to previous
  Llook at next lowest unvisited option

(E) 0 9 3 2 4 6 1 5 8 7

b. List all vertices in post-order.

Visit all children in natural order - only go 1 deep on back up

Visit self

(A) 5 8 6 2 1 4 7 3 9 0

(B) 8 7 2 5 4 1 6 3 9 0

(C) 4 1 5 6 8 7 2 3 9 0

[✓ (D)] 1 6 8 7 2 5 4 3 9 0

(E) 4 5 6 1 8 7 2 3 9 0

c. List all vertices in reverse post-order.

Visit self

Visit all children in natural order

[✓ (A)] 0 9 3 4 5 2 7 8 6 1

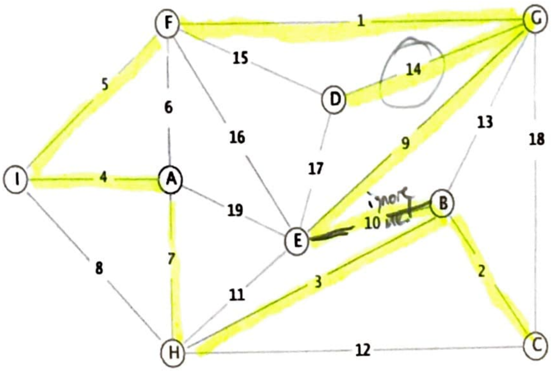
(B) 0 9 3 7 4 1 2 6 8 5

(C) 0 9 3 2 7 8 1 6 5 4

(D) 0 9 3 2 7 8 6 5 1 4

(E) 0 9 3 6 1 4 5 2 7 8

Problem 7. Consider the following edge-weighted graph with 9 vertices and 19 edges. Note that the edge weights are distinct integers between 1 and 19.



a. What is the *last* edge that is added to the minimum spanning tree (MST) by Kruskal's algorithm?

Find smallest unvisited edge
if it does not create a loop, add it
All connected nodes should be apart of graph

(A) 10

(B) 16

[✓ (C)] 14

(D) 12

(E) 8

b. What is the weight of the MST?

[✓ (A)] 45

Add weights of all edges

(B) 48

(C) 36

(D) 50

(E) 56

Problem 8. Suppose that after running Dijkstra's algorithm on an edge-weighted digraph, starting from vertex `0`, the values in the `distTo` and `edgeTo` arrays are as shown below.

v	distTo[v]	edgeTo[v]
0	0	null
1	13	6 -> 1
2	6	0 -> 2
3	18	9 -> 3
4	6	0 -> 4
5	9	4 -> 5
6	10	4 -> 6
7	9	4 -> 7
8	7	4 -> 8
9	14	6 -> 9
10	14	6 -> 10

1. Find edgeTos that start with
0 -> x

0 -> 2

0 -> 4

2. Now find edges that start with X

2: None -> eliminated

4: 4 -> 6
4 -> 5
4 -> 7

3. Edges that start with X

6: 6 -> 1
6 -> 9
6 -> 10

5: None

7: None

4. Edges that start with X

1: None

9: 9 -> 3

10: None

5. We found 3, So our path is

walk backwards to step 1

a. What is the shortest path to vertex 3?

(A) 0 -> 4 -> 6 -> 9 -> 1 -> 3

(B) 0 -> 4 -> 8 -> 1 -> 3

(C) 0 -> 10 -> 1 -> 7 -> 3

(D) 0 -> 4 -> 6 -> 9 -> 6 -> 3

[✓ (E) 0 -> 4 -> 6 -> 9 -> 3]

0->4->6->9->3

b. What is the weight on the edge `6 -> 9`?

(A) 8

(B) 2

(C) 6

[✓ (D) 4]

(E) 10

distances

$$0 \xrightarrow[6]{6} 4 \xrightarrow[4]{10} 6 \xrightarrow[4]{14} 9 \xrightarrow[4]{18} 3$$

weights

subtract previous weights from current

↳ 1st ~~[illegible]~~ distance is that edge