CSC 112: Computer Operating Systems

Midterm Exam Spring 2025

Department of Computer Science,
Hofstra University

Q1. Multiple-choice (15 pts)

1. Consider the following functions from positives integers to real numbers (√n denotes SquareRoot(n)): 10, √n, n, $\log_2 n$, 100/n.
The correct arrangement of the above functions in increasing order of asymptotic complexity is:
(A) $\log_2 n$, 100/n, 10, √n, n
(B) 100/n, 10, $\log_2 n$, √n, n
(C) 10, 100/n, √n, $\log_2 n$, n
(D) 100/n, $\log_2 n$, 10, √n, n
ANS: B

2. Consider the following three functions:
$f_1 = 10^n$
$f_2 = n^{(\log n)}$
$f_3 = n^{(\sqrt{n})}$
Which one of the following options arranges the functions in the **increasing order of asymptotic** complexity?
(A) f_3, f_2, f_1
(B) f_2, f_1, f_3
(C) f_1, f_2, f_3
(D) f_2, f_3, f_1
ANS: D

3. What is the time complexity of function f1(n) and function f2(n), respectively?

```
void f1(n){
    for (int i = 0; i < n; i+=5) {
          // O(1)
}
}
void f2(n){
    for (int i = 1; i < n; i*=5) {
          // O(1)
}
}
```

A) O(\log n), O(\log n)
B) O(\log n), O(n)
C) O(n), O(\log n)
D) O(n), O(n)
Answer: C

4. What does the regular expression pattern ^[0-9]+$ match?
A) A string containing at least one digit
B) A string starting with a digit
C) A string ending with a digit
D) A string containing only digits
Answer: A or D

5. Which of the following regular expressions matches exactly three consecutive lowercase letters?
A) [a-z]{3}
B) [a-z]{1,3}
C) [a-z]{3,}
D) [a-z]{0,3}
Answer: A

6. What does the regular expression [^abc] match?
A) Either a, b, or c
B) Any character that is not a, b, or c
C) The beginning of a string followed by a, b, or c
D) The characters 'a', 'b', and 'c' only when they appear together
Answer: B

7. Which regular expression correctly matches a valid email address format (username@domain.com)?
A) [a-zA-Z0-9]+@[a-zA-Z0-9]+.[a-zA-Z]{2,}
B) [a-zA-Z0-9]@[a-zA-Z0-9].[a-zA-Z]
C) [a-zA-Z0-9]@[a-zA-Z0-9].[a-zA-Z]*
D) .+@.+..+
Answer: A

8. Which regular expression matches a string that contains either "cat" or "dog"?
A) cat|dog
B) (cat)(dog)
C) cat+dog
D) [cat|dog]
Answer: A

9. Which of the following would match a string containing at least one digit?
A) \d+
B) \d{1,}
C) \d?
D) Both A and B
Answer: D

10. What does the regular expression \b[A-Z][a-z]*\b match?
A) Any capitalized word (a word that starts with an uppercase letter followed by zero or more lowercase letters)
B) Any word containing at least one uppercase letter in it
C) Any word written entirely in uppercase
D) Any word with exactly one uppercase letter in it
Answer: A

Q1. Multiple-choice (15 pts)

11. Primary clustering in linear probing occurs because:
A) Hash functions produce sequential indices
B) Collisions form long contiguous blocks
C) Table size is a prime number
D) Keys are not uniformly distributed
Answer: B

12. Quadratic probing uses which probe sequence?
A) h+1, h+2, h+3,...
B) $h+1^2$, $h+2^2$, $h+3^2$,...
C) h+hash2(key), 2*hash2(key),...
D) Random permutation
Answer: B

13. Which of the following is NOT a method to mitigate primary clustering?
A) Better-designed hash function
B) Alternative probing methods
C) Resizing the hash table
D) Using a binary search tree
Answer: D

14. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty Binary Search Tree. What is the in-order traversal of the resultant tree?
A) 7 5 1 0 3 2 4 6 8 9
B) 0 2 4 3 1 6 5 9 8 7
C) 0 1 2 3 4 5 6 7 8 9
D) 9 8 6 4 2 3 0 1 5 7
ANS: C

15. A full binary tree with height 3 has how many nodes?
A) 7
B) 15
C) 31
D) 8
Answer: B 15 [Formula: n=2^(h+1)−1 for h=3]

Q1 Multiple-choice questions: enter your answer keys here:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
B	D	C	A or D	A	B	A	A	D	A
11	**12**	**13**	**14**	**15**					
B	B	D	C	B					

Q2 Lecture 3-inheritance and polymorphism

- a) (1 pts) What does the main method of MyClass Tester print?
- ANS: true, since both c2 and c3 have member variables a=30 and b=29.7

```
public class MyClass {
  public int a;
  public double b;
  public MyClass(int first, double second) {
    this.a = first;
    this.b = second;
  }
  public boolean same(MyClass other) {
    return other.a == this.a && other.b == this.b;
  }
}
```

```
public class MyClassTester {
  public static void main(String[] args) {
    MyClass c1 = new MyClass(30, 123.9);
    MyClass c2 = new MyClass(30, 29.7);
    MyClass c3 = new MyClass(c1.a, c2.b);

    System.out.println(c2.same(c3));
  }
}
```

Q2 Lecture 3-inheritance and polymorphism

- b) (4 pts) Consider the following class definitions:
- b1) What does the following program print? Explain why.

```
Person u = new Undergrad();
u.method1();
```

- ANS: Infinite recursive calling of method1() to print “Student 1” repeatedly. Since the actual object type is Undergrad, but method1() is not defined in class Undergrad, so the method1() defined its parent class will be called to print “Student 1” in an infinite recursion.
- b2) What does the following program print? Explain why.

```
Person u = new Undergrad();
u.method2();
```

- ANS: It prints “Undergrad 2”. Since the actual object type is Undergrad, the method2() defined in class Undergrad will be called to print “Undergrad 2”.

```
public class Person {
  public void method1() {
    System.out.print("Person 1 ");
  }
  public void method2() {
    System.out.print("Person 2 ");
  }
}
```

```
public class Student extends Person {
  public void method1() {
    System.out.print("Student 1 ");
    method1();
    method2();
  }
  public void method2() {
    System.out.print("Student 2 ");
  }
}
```

```
public class Undergrad extends Student {
  public void method2() {
    System.out.print("Undergrad 2 ");
  }
}
```

Q3. Lecture 5-algorithm performance analysis (10 pts)

For each function $f(n)$ below, give an asymptotic upper bound using big-O notation. You should give the tightest bound possible (so giving $O(2^n)$ for every question is unlikely to result in many points).

(a) $f(n) = 1000000$
(b) $f(n) = n^4 + 100n^3 + 14n^2$
(c) $f(n) = 2^n + 100n^3 + 14n^2$
(d) $f(n) = 100n^3 + 14n^2$
(e) $f(n) = 7n^2 + 14n$
(f) $f(n) = \log(7n^2)$
(g) $f(n) = 5\log\log n + 4\log(n)*\log(n)$
(h) $f(n) = .001n + 100*2^n$
(i) $f(n) = n^3(1 + 6n + 2014n^2)$
(j) $f(n) = (\log n)(n + n^2)$

(a) $O(1)$
(b) $O(n^4)$
(c) $O(2^n)$
(d) $O(n^3)$
(e) $O(n^2)$
(f) $O(\log n)$
(g) $O(\log^2 n)$
(h) $O(2^n)$
(i) $O(n^5)$
(j) $O(n^2 \log n)$

Q4. Lecture 7-hash table (20 pts)

Insert the following six keys in this order: 19, 48, 8, 27, 97, 7 into a hash table of size 10, where the hash function is modulo table size (%10).
a) (1 pts) What is the load factor?
ANS: Load factor = number of keys / size of hash table = 6/10 = 0.6

b) (3 pts) Fill in the table, resolving hash collisions with linear probing. **(Fill in the table without detailed steps.)**
ANS:

- 19 (insert at index 9)
- 48 (insert at index 8)
- 8 (collision with 48). Insert at index 0
- 27 (insert at index 7)
- 97 (collision with 27). Insert at index 1
- 7 (collision with 27). Insert at index 2

0	1	2	3	4	5	6	7	8	9
8	97	7					27	48	19

Q4. Lecture 7-hash table (20 pts)

c) (4 pts) Fill in the table, resolving hash collisions with quadratic probing. (Show the probe sequence for resolving collisions.)

ANS:

- 19 (insert at index 9)
- 48 (insert at index 8)
- 8 (collision with 48 at index 8)
  - (8 + 1^2) % 10 = 9 (collision with 19)
  - (8 + 2^2) % 10 = 2. Insert at index 2
- 27 (insert at index 7)
- 97 (collision with 27 at index 7)
  - (7 + 1^2) % 10 = 8 (collision with 48)
  - (7 + 2^2) % 10 = 1 (empty). Insert at index 1
- 7 (collision with 27 at index 7)
  - (7 + 1^2) % 10 = 8 (collision with 48)
  - (7 + 2^2) % 10 = 1 (collision with 97)
  - (7 + 3^2) % 10 = 6 (empty). Insert at index 6

0	1	2	3	4	5	6	7	8	9
	97	8				7	27	48	19

Q4. Lecture 7-hash table (20 pts)

d) (8 pts) Fill in the table, resolving hash collisions with double hashing with two hash functions: h1(k)=x%10 (primary hash), h2(k)=1+(x%7) (secondary hash), Probing formula: Probe(k, i)=(h1(k)+i·h2(k))%10, i=0, 1, 2… **(Show the probe sequence for resolving collisions.)**

- 19 (insert at index 9)
- 48 (insert at index 8)
- 8 (collision with 48 at index 8)
  - $h_2(8)=1+(8\%7)=2$. Next probe: $(8 + 1\cdot2)\%10 = 0$ (insert at index 0)
- 27 (insert at index 7)
- 97 (collision with 27 at index 7)
  - $h_2(97)=1+(97\%7)=7$. Next probe: $(7 + 1\cdot7)\%10 = 4$ (insert at index 4)
- 7 (collision with 27 at index 7)
  - $h_2(7)=1+(7\%7)=1$. Next probes:
  - $i=1$: $(7 + 1\cdot1)\%10 = 8$ (collision with 48)
  - $i=2$: $(7 + 2\cdot1)\%10 = 9$ (collision with 19)
  - $i=3$: $(7 + 3\cdot1)\%10 = 0$ (collision with 8)
  - $i=4$: $(7 + 4\cdot1)\%10 = 1$ (insert at index 1)

0	1	2	3	4	5	6	7	8	9
8	7			97			27	48	19

Q4. Lecture 7-hash table (20 pts) (with typo)

(The exam paper had a typo Probe(k, i)=(h1(k)+i·h2(k))%7. Despite this typo, it is still a valid hash function and you can still solve this problem based on this probe function. Students who give either solution is given full credit.)

d) (8 pts) Fill in the table, resolving hash collisions with double hashing with two hash functions: h1(k)=x%10 (primary hash), h2(k)=1+(x%7) (secondary hash), Probing formula: Probe(k, i)=(h1(k)+i·h2(k))%7, i=0, 1, 2… **(Show the probe sequence for resolving collisions.)**

- 19 (insert at index 9)
- 48 (insert at index 8)
- 8 (collision with 48 at index 8)
  - $h_2(8)=1+(8\%7)=2$. Next probe: $(8 + 1\cdot2)\%7 = 3$ (insert at index 3)
- 27 (insert at index 7)
- 97 (collision with 27 at index 7)
  - $h_2(97)=1+(97\%7)=7$. Next probe: $(7 + 1\cdot7)\%7 = 0$ (insert at index 0)
- 7 (collision with 27 at index 7)
  - $h_2(7)=1+(7\%7)=1$. Next probe: $(7 + 1\cdot1)\%7 = 1$ (insert at index 1)

0	1	2	3	4	5	6	7	8	9
97	7		8				27	48	19

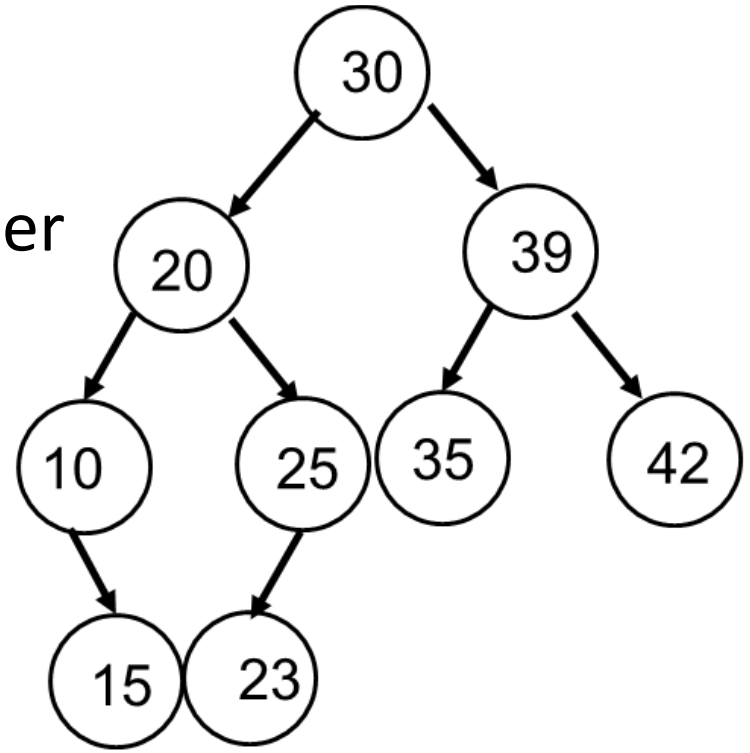
Q4: Separate Chaining

- e) (2 pts) Fill in the table, resolving hash collisions with separate chaining into a sorted linked list (with the smallest element at the head of the list). (Fill in the table without detailed steps.)
- ANS: Upon hash collision, put the item into a sorted linked list in the index.

0	1	2	3	4	5	6	7	8	9
							7,27,97	8,48	19

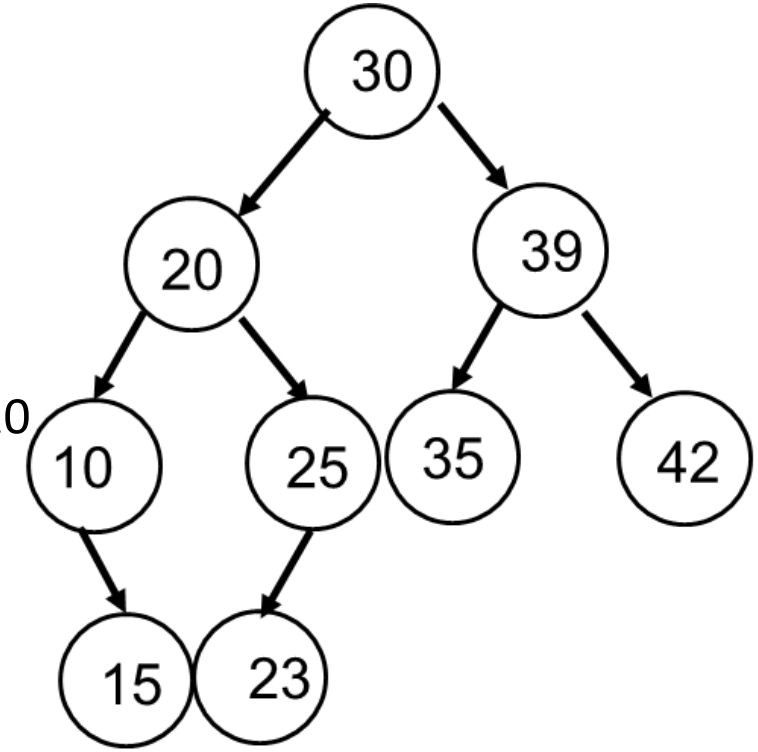
Q5. Lecture 8-binary search tree (40 pts)

- a) (15 pts) The Pre-order traversal sequence of a Binary Search Tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Construct the tree and give its in-order and post-order traversals. Show the detailed steps.
- ANS: Given:
- Pre-order: 30 20 10 15 25 23 39 35 42
- In-order: 10 15 20 23 25 30 35 39 42
- We can construct the tree and derive the post-order
- Post-order: 15 10 23 25 20 35 42 39 30



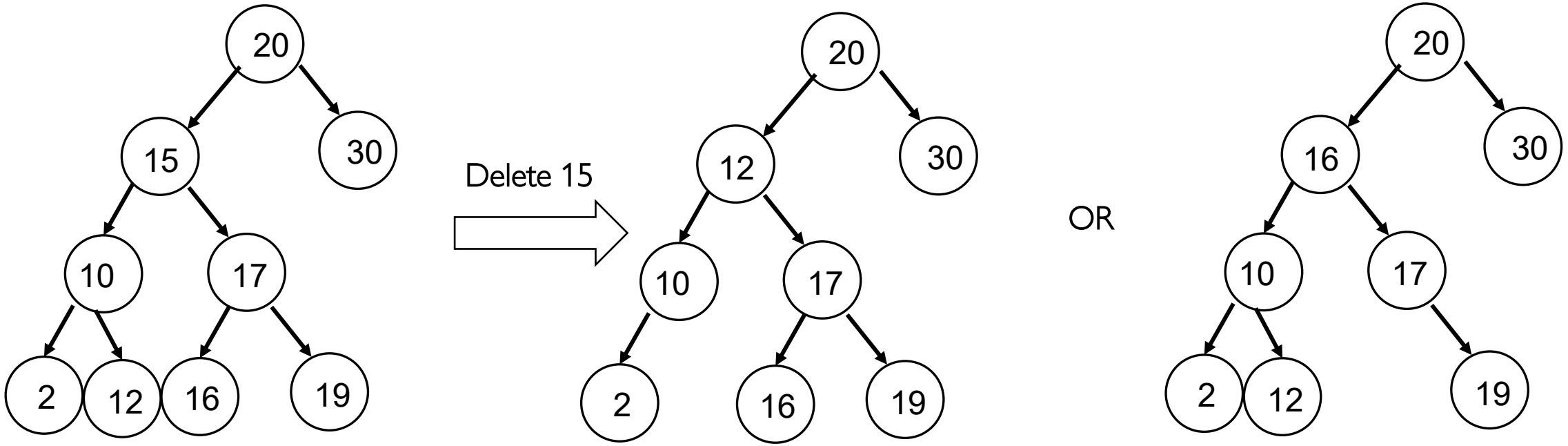
Q5. Lecture 8-binary search tree (40 pts)

- a) Explanations:
- Pre-order: 30 20 10 15 25 23 39 35 42
- In-order: 10 15 20 23 25 30 35 39 42
- 1. Root Identification: The first element in pre-order (30) is the root.
- 2. Subtree Division:
- - In in-order traversal, elements before 30 (10 15 20 23 25) form the left subtree, and elements after (35 39 42) form the right subtree.
- 3. Left Subtree Construction:
- - Pre-order segment for left: 20 10 15 25 23
- - Root is 20 (first in pre-order). In in-order (10 15 20 23 25), left of 20 is (10 15) and right is (23 25).
- - Pre-order segment (10 15): root is 10, and 15 is its right child
- - Pre-order segment (25 23): root is 25, and 23 is its left child
- 4. Right Subtree Construction:
- - Pre-order segment for right: (39 35 42)
- - Root is 39 (first in pre-order). In in-order (35 39 42), 39 has a left child 35 and right child 42.
- We can draw the tree now and derive the post order traversal 15 10 23 25 20 35 42 39 30.



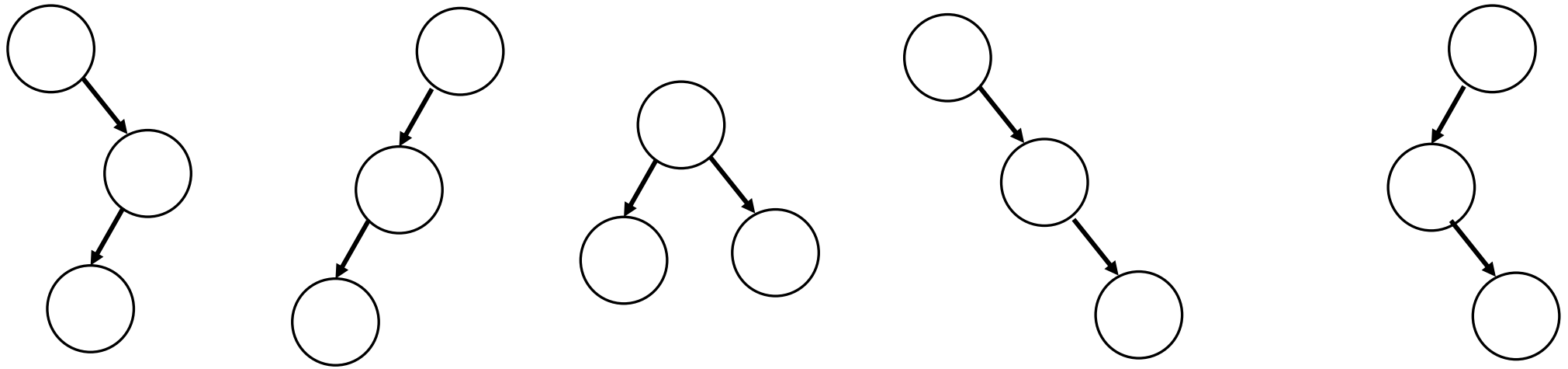
Q5. Lecture 8-binary search tree (40 pts)

- b) (5 pts) Given this Binary Search Tree, draw the resulting Binary Search Tree after deleting node 15. There are two possible results, and please draw both of them.

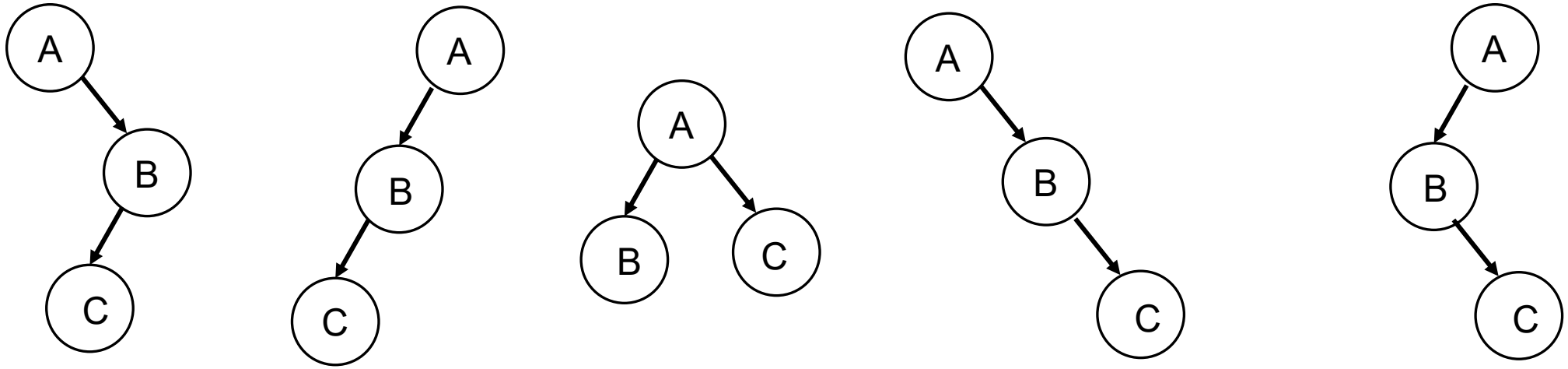


Q5. Lecture 8-binary search tree (40 pts)

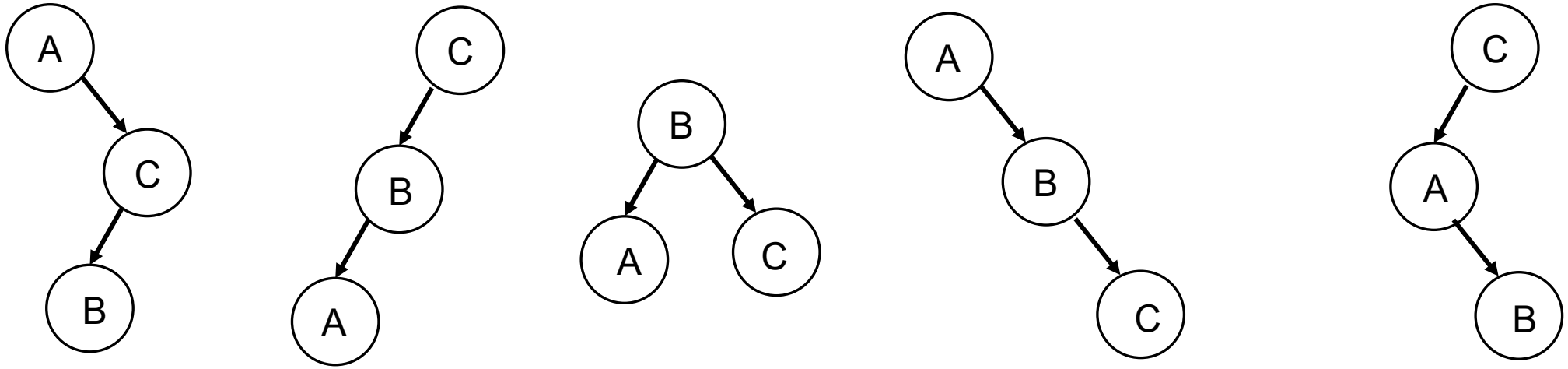
- c) (20 pts) Fill in the labels A, B, C in each node in the following Binary Trees so they all have:
- either (1) pre-order traversal of ABC
- or (2) In-order traversal of ABC
- or (3) post-order traversal of ABC
- or (4) they are all Binary Search Trees, considering A < B < C



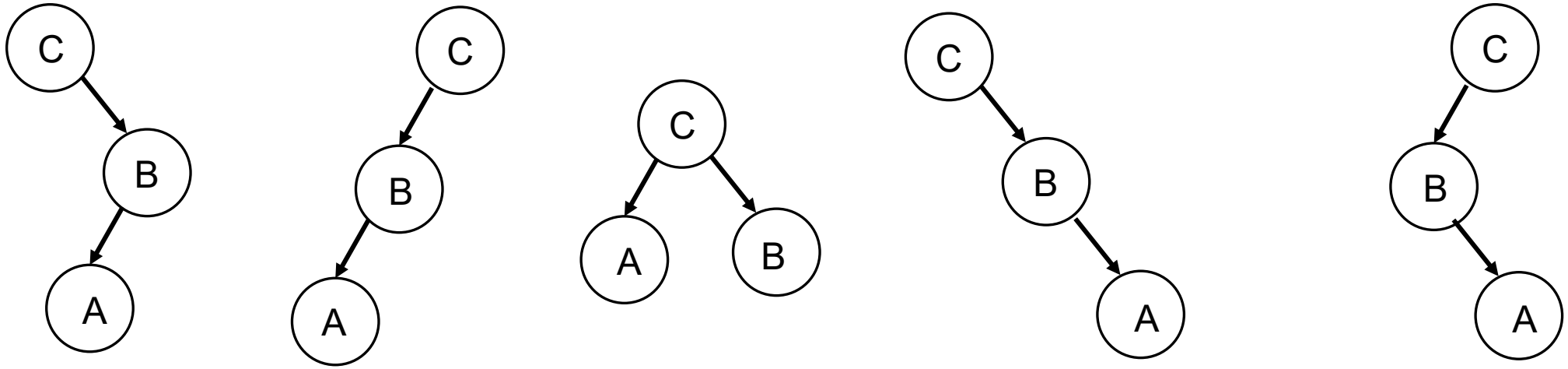
(1) pre-order traversal of ABC



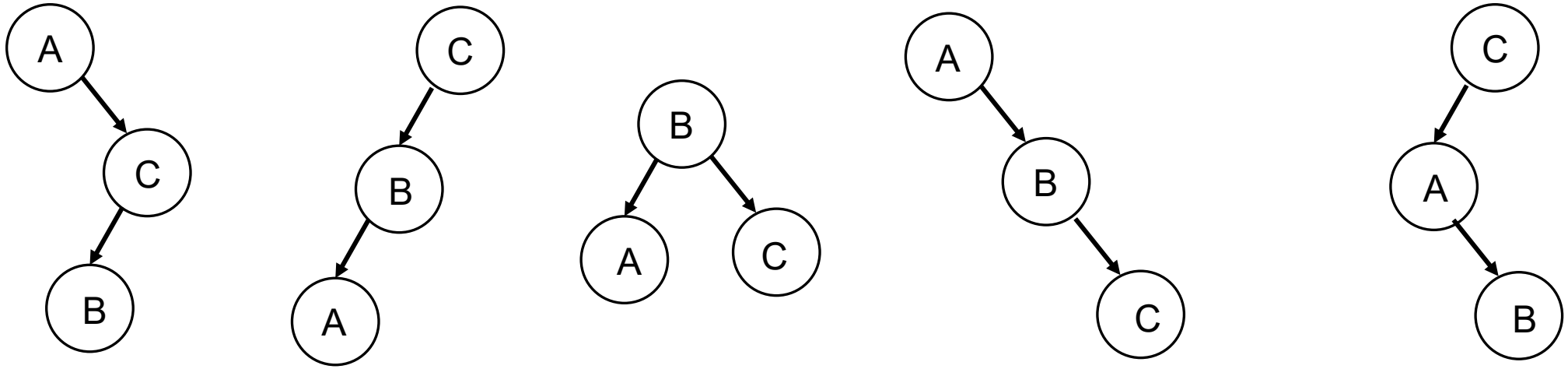
(2) In-order traversal of ABC



(3) post-order traversal of ABC



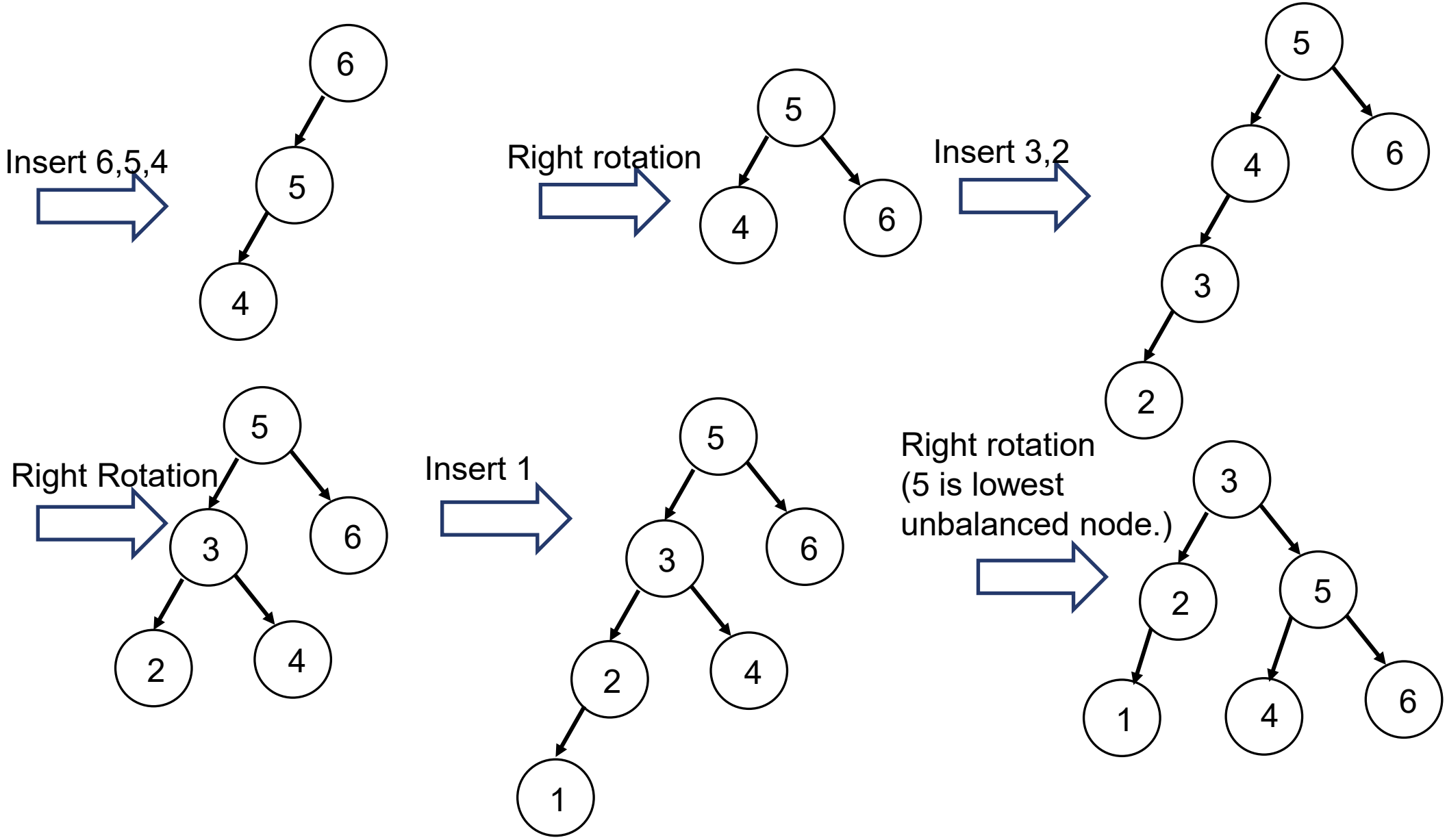
(4) they are all Binary Search Trees, considering A < B < C



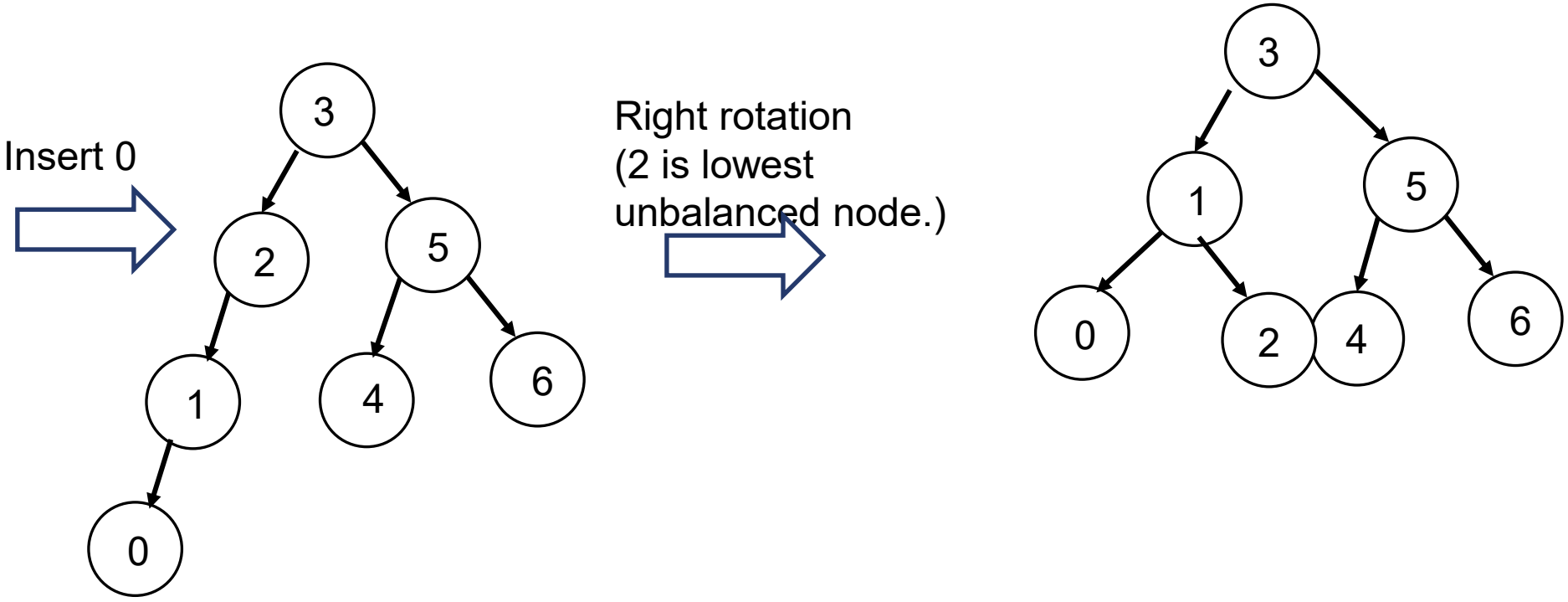
Q6 Lecture 9-self balancing trees (20 pts)

- a) (10 pts) Create an AVL Tree by inserting the sequence: 6, 5, 4, 3, 2, 1, 0. Draw a new figure whenever you do a rotation. (Do not write out the AVL invariant at each step, just draw the tree at each step.) (If you run out of space, use back of the paper.)

AVL Tree: Inserting the sequence: 6, 5, 4, 3, 2, 1, 0



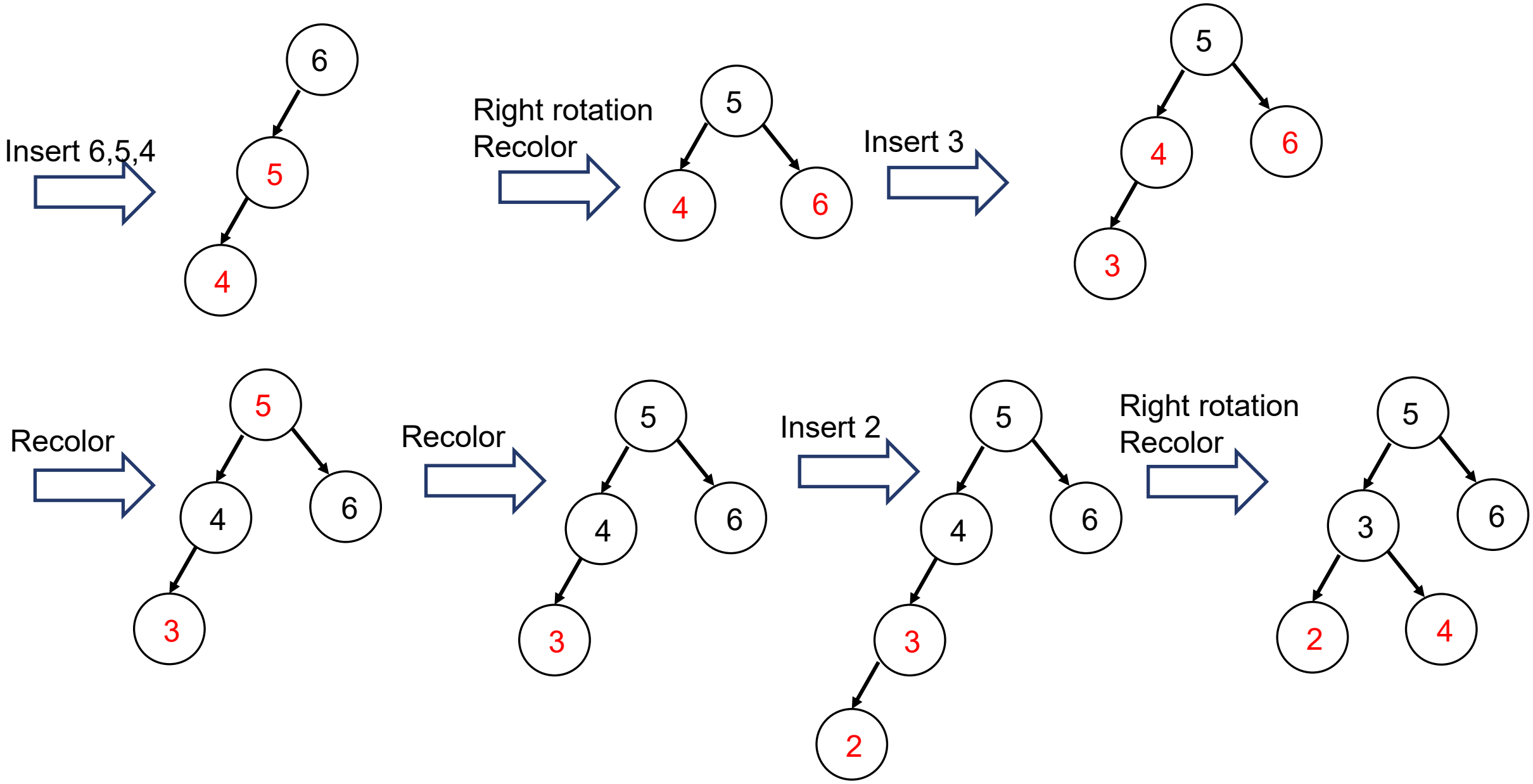
AVL Tree: Inserting the sequence: 6, 5, 4, 3, 2, 1, 0



Q6 Lecture 9-self balancing trees (20 pts)

- b) (10 pts) Create a Red-Black Tree by inserting the sequence: 6, 5, 4, 3, 2, 1, 0. Draw a new figure whenever you do a rotation and/or recoloring. (If you run out of space, use back of the paper.)

Red-Black Tree: Inserting the sequence: 6, 5, 4, 3, 2, 1, 0



Red-Black Tree: Inserting the sequence: 6, 5, 4, 3, 2, 1, 0