CSC017 Spring 2025 Midterm Exam

Student Name：_____ID：

Total Points	

Note: This exam paper should be kept confidential and any dissemination violates copyright.

Q1	Q2	Q3	Q4	Q5	Q6
/15	/5	/10	/20	/40	/20

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

1	2	3	4	5	6	7	8	9	10
11	**12**	**13**	**14**	**15**					

Q1. Multiple-choice. (15 pts)
For the following multiple-choice questions, each question has exactly one correct answer key. If multiple choices are correct, choose the option "All of the above". **Fill in the answer keys in the table above. (Answer keys written in the question area will not be counted.)**

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

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

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)

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

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}

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

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) .+@.+..+

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]

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

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

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

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

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

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

15. A full binary tree with height 3 has how many nodes?
A) 7
B) 15
C) 31
D) 8

Q2 Lecture 3-inheritance and polymorphism (5 pts)
a) (1 pts) What does the main method of MyClass Tester print?

```
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));
  }
}
```

ANS:

b) (4 pts) Consider the following class definitions:

```
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 ");
  }
}
```

b1) What does the following program print? Explain why.

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

ANS:

b2) What does the following program print? Explain why.

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

ANS:

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) $f(n) = (\log n)(n + n^2)$

ANS:

(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) $f(n) = (\log n)(n + n^2)$ _____

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:

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

0	1	2	3	4	5	6	7	8	9

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

0	1	2	3	4	5	6	7	8	9

d) (10 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.)
ANS:

0	1	2	3	4	5	6	7	8	9

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:

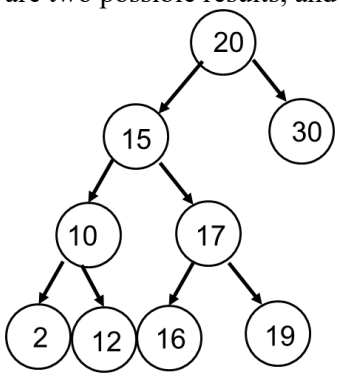
0	1	2	3	4	5		6	7	8	9

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.

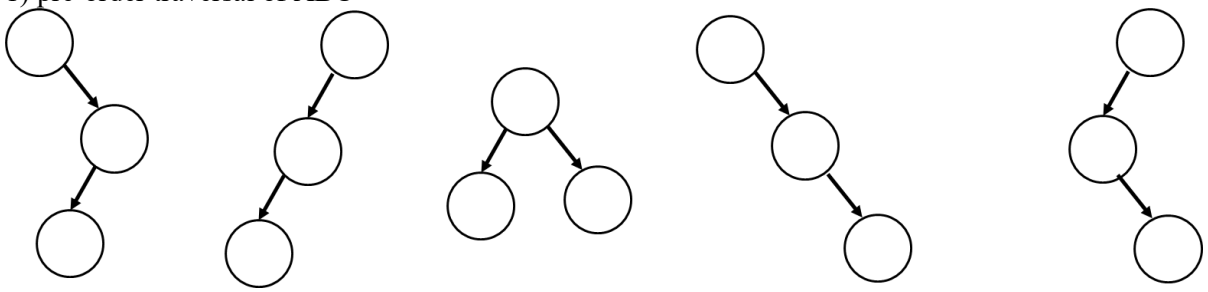
ANS:

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.

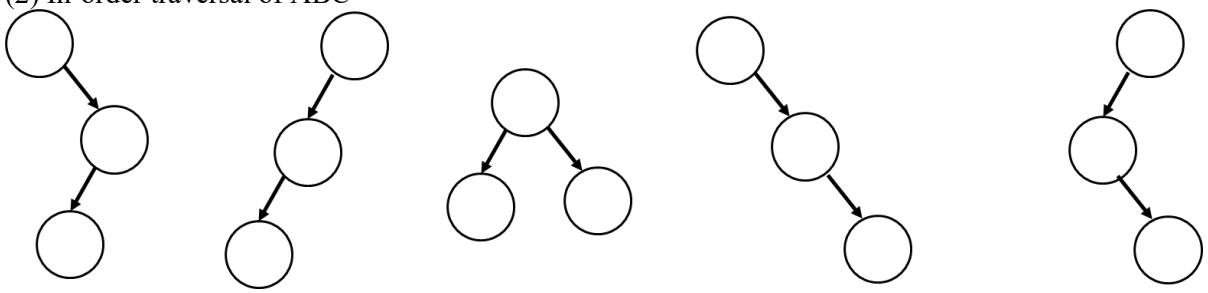


c) (20 pts) Fill in the labels A, B, C in each node in the following Binary Trees so they all have:
either (1) pre-order traversal of ABC
or (2) In-order traversal of ABC
or (3) post-order traversal of ABC
or (4) they are all Binary Search Trees, considering A < B < C

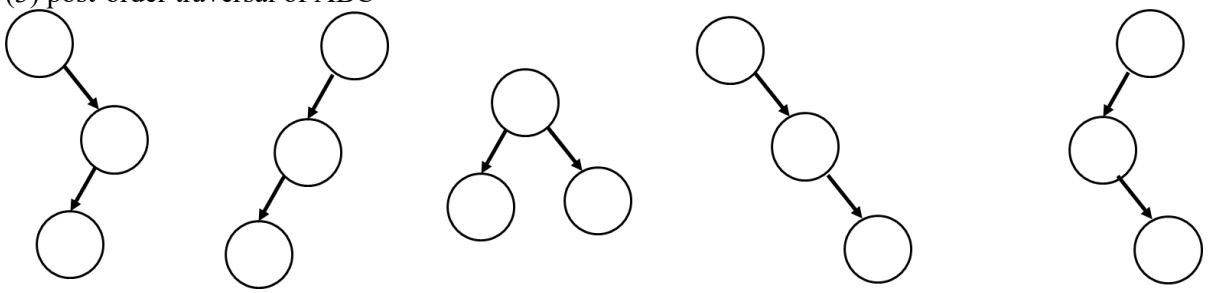
1) pre-order traversal of ABC



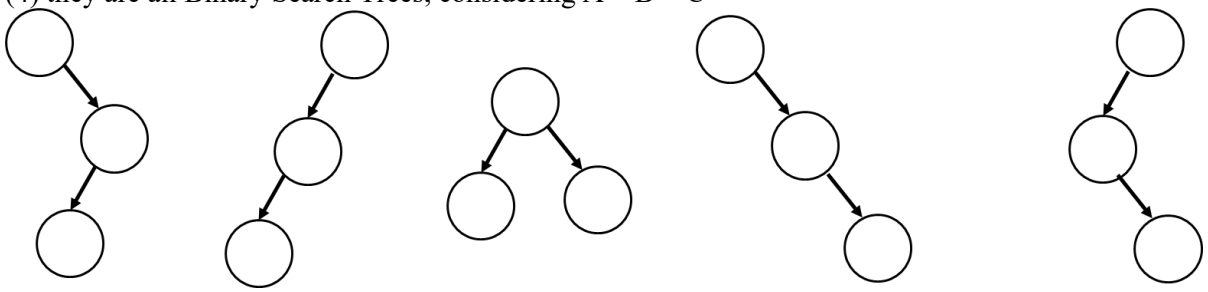
(2) In-order traversal of ABC



(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.)

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.)