

CS 142 Sample Final Exam Key #2

1. Big-Oh Analysis

- a) $O(N^4)$
- b) $O((\log N)^2)$
- c) $O(N)$
- d) $O(N^2)$
- e) $O(N^2)$
- f) $O(N)$

2. 2D Array Mystery

```
{0, 0, -24, 98, 13, 1},  
{2, 0, 0, 0, 1, 1},  
{0, 50, -1, 4, -1, 5},  
{3, 49, 13, 4, 13, 52},  
{14, 49, 49, 4, 14, 4}}
```

3. Recursion Mystery

Call	Output
mystery(3, 3);	=3=
mystery(5, 1);	*
mystery(1, 5);	5 4 =3= 2 1
mystery(2, 7);	7 6 5 * 4 3 2
mystery(1, 8);	8 7 6 5 * 4 3 2 1

4. Recursion Programming

```
public static String repeat(String s, int n) {  
    if (n < 0) {  
        throw new IllegalArgumentException();  
    } else if (n == 0) {  
        return "";  
    } else if (n == 1) {  
        return s;  
    } else if (n % 2 == 0) {  
        String temp = repeat(s, n / 2);  
        return temp + temp;  
        // alternative without temp: return repeat(s + s, n / 2);  
    } else {  
        return s + repeat(s, n - 1);  
    }  
}
```

5. Inheritance and Comparable Programming

```
public class CalendarDate extends Date implements Comparable<CalendarDate> {  
    private int year;  
  
    public CalendarDate(int year, int month, int day) {  
        super(month, day);  
        this.year = year;  
    }  
}
```

```

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

public String toString() {
    return year + "/" + super.toString();
}

public void nextDay() {
    super.nextDay();
    if (getMonth() == 1 && getDay() == 1) {
        year++; // wrap year
    }
}

public int compareTo(CalendarDate other) {
    if (year != other.year) {
        return year - other.year;
    } else if (getMonth() != other.getMonth()) {
        return getMonth() - other.getMonth();
    } else {
        return getDay() - other.getDay();
    }
}
}

```

6. Searching and Sorting

(a) Indexes examined: 7, 11, 13, 12 Value returned: -14

(b) {-9, 5, 8, 14, 0, -1, -7, 3}
 {-9, -7, 8, 14, 0, -1, 5, 3}
 {-9, -7, -1, 14, 0, 8, 5, 3}

(c) { 8, 5, -9, 14, 0, -1, -7, 3}
 { 8, 5, -9, 14} { 0, -1, -7, 3} split
 { 8, 5} {-9, 14} { 0, -1} {-7, 3} split
 { 8} { 5} {-9} {14} { 0} {-1} {-7} { 3} split
 { 5, 8} {-9, 14} {-1, 0} {-7, 3} merge
 {-9, 5, 8, 14} {-7, -1, 0, 3} merge
 {-9, -7, -1, 0, 3, 5, 8, 14} merge

7. 2D Array Programming

```

public static int numUnique(int[][] lis) {
    List<Integer> unique = new ArrayList<>();
    for(int i = 0; i < lis.length; i++) {
        for(int j = 0; j < lis[i].length) {
            if(!unique.contains(lis[i][j])) {
                unique.add(lis[i][j]);
            }
        }
    }
    return unique.size();
}
}

```

}S