

CS 142 Sample Final Exam Key #1

1. Big-Oh Analysis

- a) $O(N^3)$
- b) $O(N^2)$
- c) $O(N)$
- d) $O(\log N)$
- e) $O(N^2)$
- f) $O(N^2)$

2. Array of Arrays

```
{ {2, 5, 2, 34, 23},  
  {14, 2, 6, 9, 3},  
  {54, 8, 35, 22, 44},  
  {17, 4, 61, 7, 23},  
  {1, 123, 0, 77, 32} }
```

3. Recursion Mystery

Call	Value Returned
mystery(7)	8
mystery(42)	53
mystery(385)	496
mystery(-790)	-801
mystery(89294)	90305

4. Recursion Programming

```
public static int digitMatch(int x, int y) {  
    if (x < 0 || y < 0) {  
        throw new IllegalArgumentException();  
    } else if (x < 10 || y < 10) {  
        if (x % 10 == y % 10) {  
            return 1;  
        } else {  
            return 0;  
        }  
    } else if (x % 10 == y % 10) {  
        return 1 + digitMatch(x / 10, y / 10);  
    } else {  
        return digitMatch(x / 10, y / 10);  
    }  
}
```

5. Inheritance and Comparable Programming

```
public class Point3D extends Point implements Comparable<Point3D> {  
    private int z;  
  
    public Point3D() {  
        this(0, 0, 0);  
    }  
  
    public Point3D(int x, int y, int z) {
```

```

        // super(x, y);
        // this.z = z;
        setLocation(x, y, z);
    }

    public int getZ() {
        return z;
    }

    public String toString() {
        return "(" + getX() + ", " + getY() + ", " + z + ")";
    }

    public void setLocation(int x, int y) {
        setLocation(x, y, 0);
    }

    public void setLocation(int x, int y, int z) {
        super.setLocation(x, y);
        this.z = z;
    }

    public double distanceFromOrigin() {
        return Math.sqrt(getX() * getX() + getY() * getY() + z * z);
    }

    public int compareTo(Point3D other) {
        if (getX() != other.getX()) {
            return getX() - other.getX();
        } else if (getY() != other.getY()) {
            return getY() - other.getY();
        } else {
            return z - other.z;
        }
    }
}

```

6. Searching and Sorting

- (a) Indexes examined: 7, 3, 5 Value returned: 5
- (b) {12, 29, 19, 48, 23, 55, 74, 37}
 {12, **19**, **29**, 48, 23, 55, 74, 37}
 {12, 19, **23**, 48, **29**, 55, 74, 37}
- (c) {37, 29, 19, 48, 23, 55, 74, 12}
 {37, 29, 19, 48} {23, 55, 74, 12} **split**
 {37, 29} {19, 48} {23, 55} {74, 12} **split**
 {37} {29} {19} {48} {23} {55} {74} {12} **split**
 {29, 37} {19, 48} {23, 55} {12, 74} **merge**
 {19, 29, 37, 48} {12, 23, 55, 74} **merge**
{12, 19, 23, 29, 37, 48, 55, 74} **merge**

7. 2D Array Programming

```

public static int findMax(int[][] lis) {
    int maxSum = 0;
    int maxRow = 0;
    for(int i = 0; i < lis.length; i++) {
        int curSum = 0;
        int curRow = i;
        for(int j = 0; j < lis[i].length; j++) {
            curSum += lis[i][j];
        }
    }
}

```

```
    }
    if(curSum > maxSum) {
        maxSum = curSum;
        maxRow = curRow;
    }
}
return maxRow;
}
```