

Recursive Algorithms in Action

Binary Search Recursive

```
public static int binarySearchRecursive(int[] a, int start, int end, int target) {
    if(end < start)
        return -1;
    int middle = (start + end) / 2 ;
    if(target==a[middle])
        return middle;
    else if(target<a[middle])
        return binarySearchRecursive(a, start, middle - 1, target);
    else
        return binarySearchRecursive(a, middle + 1, end, target);
}
```

To Test

```
int[] A = {1,5,6,7,10,12,15,22,25,30,50,52,58,59,61,64,70,90,100};
int position = binarySearchRecursive(A, 0, A.length-1, 22);
```

Flood Fill

```
public void floodFill(int col, int row) {
    if (grid[col][row]==0){
        grid[col][row]=1;
        floodFill(col, row-1);
        floodFill(col, row+1);
        floodFill(col+1, row);
        floodFill(col-1, row);
    }
}
```

To Test

```
floodFil(150,150);
```

Fractal Squares

```
public void drawSquare(Graphics g, int x, int y, int s, Color c){
    g.setColor(c);
    g.fillRect(x, y, s, s);
    try { Thread.sleep(3); }
    catch (Exception e) {}
    if (s>=4){
        s = s/2;
        c = new Color( R.nextInt(255), R.nextInt(255), R.nextInt(255) );
        drawSquare(g, x-s, y-s, s, c); //draw half size square in upper left corner
        drawSquare(g, x+s*2, y-s, s, c); //draw half size square in upper right corner
        drawSquare(g, x+s*2, y+s*2, s, c); //draw half size square in lower right corner
        drawSquare(g, x-s, y+s*2, s, c); //draw half size square in lower left corner
    }
}
```

To Test: drawSquare(g, 400,400, 128, Color.Yellow);