

Advanced Kenya

Array Traversal and Tangents

Tristan Allwood - tora@doc.ic.ac.uk

Lexicographic Traversal

```
void main() {  
    int[][] array = new int[10][10];  
    int count = 0;  
    for( int i = 0 ; i < array.length ; i++ ){  
        for( int j = 0 ; j < array[i].length ; j++ ) {  
            array[i][j] = count;  
            count ++;  
        }  
    }  
}
```

1	2	3
4	5	6
7	8	9

Pascal Triangle

1							
1	1						
1	2	1					
1	3	3	1				
1	4	6	4	1			
1	5	10	10	5	1		
1	6	15	20	15	6	1	

Pascal Triangle

```
int[][] buildPascal(int size) {
    int[][] out = new int[size][];
    for(int i = 0 ; i < out.length ; i++ ) {
        out[i] = new int[i+1];
    }
    out[0][0] = 1;
    for(int i = 1 ; i < out.length ; i++ ) {
        for(int j = 0 ; j < out[i].length ; j++ ) {
            int above;
            if( j >= i ) {
                above = 0;
            } else {
                above = out[i-1][j];
            }
            int left;
            if( j == 0 ) {
                left = 0;
            } else {
                left = out[i-1][j-1];
            }
            out[i][j] = above + left;
        }
    }
    return out;
}
```

```
void main() {
    int[][] triangle = buildPascal(7);
    printArray(triangle);
}
```

```
void printArray(int[][] array) {
    for(int i = 0 ; i < array.length ; i++ ){
        for(int j = 0 ; j < array[i].length ; j++ ) {
            print(array[i][j]);
            print(' ');
        }
        println();
    }
}
```

Pascal Triangle II

1	1	1	1	1	1	1
1	2	3	4	5	6	
1	3	6	10	15		
1	4	10	20			
1	5	15				
1	6					
1						

Pascal II

```
int[][] buildPascal(int size) {
    int[][] out = new int[size][];

    for(int i = 0 ; i < size ; i++ ) {
        out[i] = new int[size-i];
    }

    for(int j = 0 ; j < size ; j++ ) {
        out[0][j] = 1;
    }

    for(int i = 1; i < out.length ; i++ ) {
        for(int j = 0; j < out[i].length ; j++ ) {
            int above = out[i-1][j];
            int left;
            if( j == 0 ) {
                left = 0;
            } else {
                left = out[i][j-1];
            }
            out[i][j] = above + left;
        }
    }
    return out;
}
```

Pascal Triangle III

1	1	1	1	1	1	1
1	2	3	4	5	6	
1	3	6	10	15		
1	4	10	20			
1	5	15				
1	6					
1						

```

int[][] buildPascal(int size) {
    int[][] out = new int[size][];

    for(int i = 0 ; i < size ; i++ ) {
        out[i] = new int[size-i];
    }

    out[0][0] = 1;

    for(int i = 1; i < out.length ; i++ ) {
        for(int j = 0; j <= i ; j++ ) {
            int y = i-j;
            int x = j;
            println(" " + x + " " + y);
            int above;
            if( y <= 0 ) {
                above = 0;
            } else {
                above = out[y-1][x];
            }
            int left;
            if(x <= 0) {
                left = 0;
            } else {
                left = out[y][x-1];
            }

            out[x][y] = above + left;
        }
    }
    return out;
}

```

Pascal III

Python Challenge

- <http://www.pythonchallenge.com/>
- Some of these are really, really evil!
- A good way to force yourself to learn any new language
- You can't solve much of this with Kenya
 - (But you could with Java...)

$$2^{38}$$

K → M
O → Q
E → G

Walk Around

Walk Around

```
class IntPair {
    int x;
    int y;
}

void main() {
    int[][] spiralled = buildSpiralArray(10,10);
    printArray(spiralled);
}

int[][] buildSpiralArray(int width, int height) {
    int[][] out = new int[width][height];
    for(int n=0 ; n < (width * height) ; n++ ) {
        IntPair coords = translate(n, width, height);
        out[coords.x][coords.y] = n;
    }

    return out;
}
```

```
IntPair translate(int n, int width, int height) {
    assert false : "TODO";
    return null;
}

void printArray(int[][] array) {
    for(int i = 0 ; i < array.length ; i++ ) {
        for(int j = 0 ; j < array[i].length ; j++ ) {
            print(array[i][j]);
            print(' ');
        }
        println();
    }
}
```

Walk Around I - Dumb

- Model a particle walking around the grid.
- Keep track of where in the grid it has been already.

1 →	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

```

IntPair translate(int n, int width, int height) {
    boolean[][] visited = new boolean[width][height];
    IntPair pos;
    int xv = 0;
    int yv = 1;

    for(int i = 0 ; i < n ; i++ ) {
        boolean cont = true;
        while(cont) {
            cont = false;
            if( xv == 1 ) {
                if(pos.x == width - 1 ||
                    visited[pos.x + xv][pos.y]) {
                    xv = 0;
                    yv = 1;
                    cont = true;
                }
            } else if ( xv == -1 ) {
                if( pos.x == 0 ||
                    visited[pos.x + xv][pos.y]) {
                    xv = 0;
                    yv = -1;
                    cont = true;
                }
            }
        }
    } else if ( yv == 1 ) {
        if( pos.y == height - 1 ||
            visited[pos.x][pos.y + yv]) {
            yv = 0;
            xv = -1;
            cont = true;
        }
    } else if ( yv == -1 ) {
        if( pos.y == 0 ||
            visited[pos.x][pos.y + yv]) {
            yv = 0;
            xv = 1;
            cont = true;
        }
    }
    visited[pos.x][pos.y] = true;
    pos.x = pos.x + xv;
    pos.y = pos.y + yv;
}
return pos;
}

```

Other Challenge Sites

- www.rubyquiz.com
- <http://perl.plover.com/qotw/>
- <http://www.javapuzzlers.com/>
- <http://www.notpron.com/>
- Or just ask google...
 - <http://www.google.co.uk/search?q=programming+challenge>