Java Programming Assignment

Rishi Dua, 2010EE50557

12 January, 2014

1 REVERSE SPIRAL

Write a program in C/C++/Java to solve following two problems:

1.1 PROBLEM STATEMENT

To print matrix (M) elements in reverse spiral order i.e., starting from the centre element, print all the elements in spiral order until the first element M[0][0] is reached. Matrix M is of order n where n is odd

1.2 ABSTRACT

Traversal (also known as tree search) refers to the process of visiting (examining and/or updating) each node in a tree data structure, exactly once, in a systematic way. The problem aims at traversing a matrix in reverse spiral order.

The java code aims at traversing through all elements of a matrix in a reverse spiral form. The problem involves using loops and recursion to match two strings containing wildcard characters '?' and '*' . '?' denotes no or exactly one character and '*' denotes no or many characters.

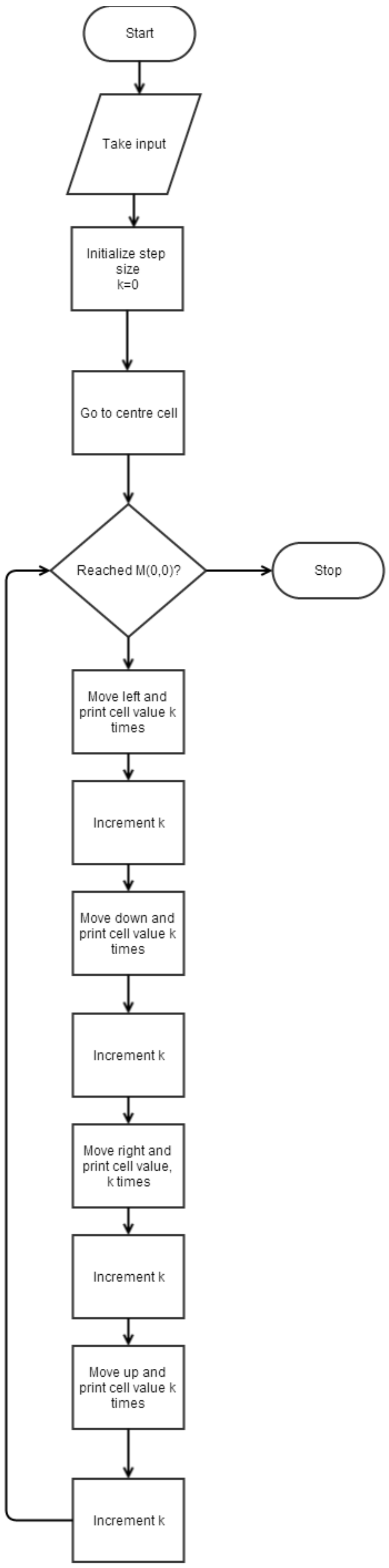
1.3 SPECIFICATION AND ASSUMPTIONS

Specifications:
Language used: Java
Platform: Ubuntu 12.04

Libraries used: java.io.BufferedReader
IDE: Eclipse

Assumptions:
n is odd

1.4 Flow chart


Start
Take input
Initialize step size k=0
Go to centre cell
Reached M(0,0)?
Stop
Move left and print cell value k times
Increment k
Move down and print cell value k times
Increment k
Move right and print cell value, k times
Increment k
Move up and print cell value k times
Increment k

1.5 Logic Implementation

Start from the centremost element.
Move in the following order: left, down, right and top increasing the number of steps moved by one every time.

1.6 Execution Directive

```
java ReverseSpiral
```

1.7 Output Of The Program

```
Enter value for n:

5
5 by 5 matrix

Enter values for following:

(0,0)
1
(0,1)
2
(0,2)
3
(0,3)
4
(0,4)
5
(1,0)
6
(1,1)
7
(1,2)
8
(1,3)
9
(1,4)
10
(2,0)
11
(2,1)
12
(2,2)
```

```
13
(2,3)
14
(2,4)
15
(3,0)
16
(3,1)
17
(3,2)
18
(3,3)
19
(3,4)
20
(4,0)
21
(4,1)
22
(4,2)
23
(4,3)
24
(4,4)
25
```

Output:

```
13
12
17
18
19
14
9
8
7
6
11
16
21
22
23
24
25
20
```

```
15
10
5
4
3
2
1
```

1.8 Result

The program prints matrix (M) elements in reverse spiral order i.e., starting from the centre element, print all the elements in spiral order until the first element M[0][0] is reached. Matrix M is of order n where n is odd

1.9 Conclusion

Successfully traversed through the elements of a matrix in reverse-spiral order as required.

2 String Matching

2.1 Problem statement

To match two strings containing wildcard characters '?' and '*' . '?' denotes no or exactly one character and '*' denotes no or many characters.

2.2 Abstract

A regular expression (abbreviated regex or regexp) is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings. The problem aims at matching two strings containing wildcards.

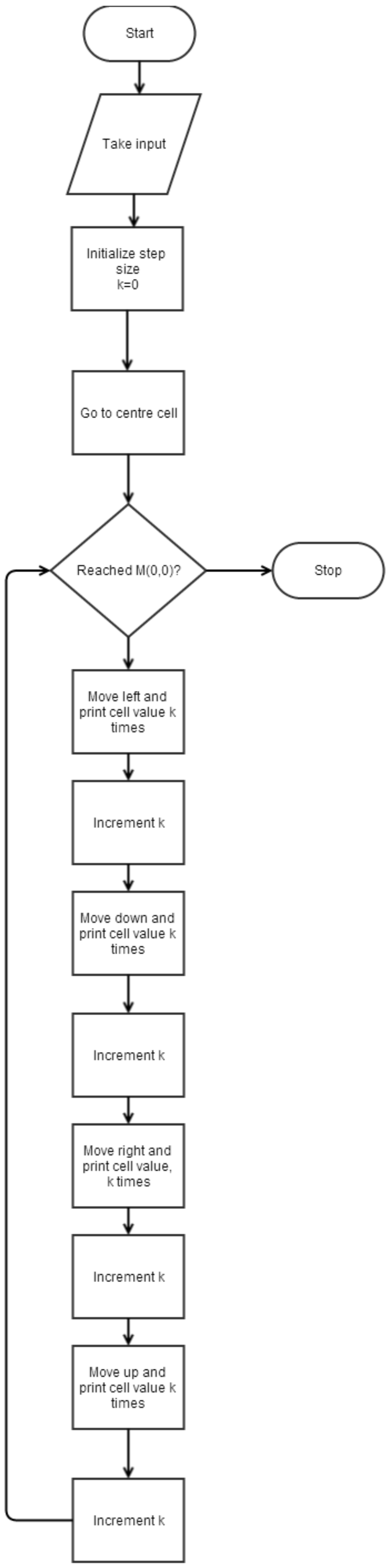
The problem is solved using Java. The problem involves using loops and recursion to match two strings containing wildcard characters '?' and '*' . '?' denotes no or exactly one character and '*' denotes no or many characters.

2.3 Specification And Assumptions

Specifications:
Language used: Java
Platform: Ubuntu 12.04
Libraries used: java.io.BufferedReader
IDE: Eclipse

Assumptions:
Only * and ? are treated as wildcards

2.4 Flow chart


Start
Take input
Initialize step size
k=0
Go to centre cell
Reached M(0,0)?
Stop
Move left and print cell value k times
Increment k
Move down and print cell value k times
Increment k
Move right and print cell value, k times
Increment k
Move up and print cell value k times
Increment k

2.5 Logic Implementation

Recursive calling is used
First of all, trivial cases are checked
If the first character is a ?, the remaining string is matched with the other string as it is and also with the first character removed If the first character is a ?, the remaining string is matched with the other string as it is and also with all the permutations beginning with the next non-wildcard character of the first string

2.6 Execution Directive

```
java RegexMatcher
```

2.7 Output Of The Program

```
Enter string 1
hello
Enter string 2
h*llo?
result: the strings match
```

2.8 Result

The program can do a simple regex comparison of two given strings.
Normal regex comparison works when both the strings contain wildcards. This code works even then.

2.9 Conclusion

Successfully compared two strings, when one/both/none contain ? and * wildcard characters.