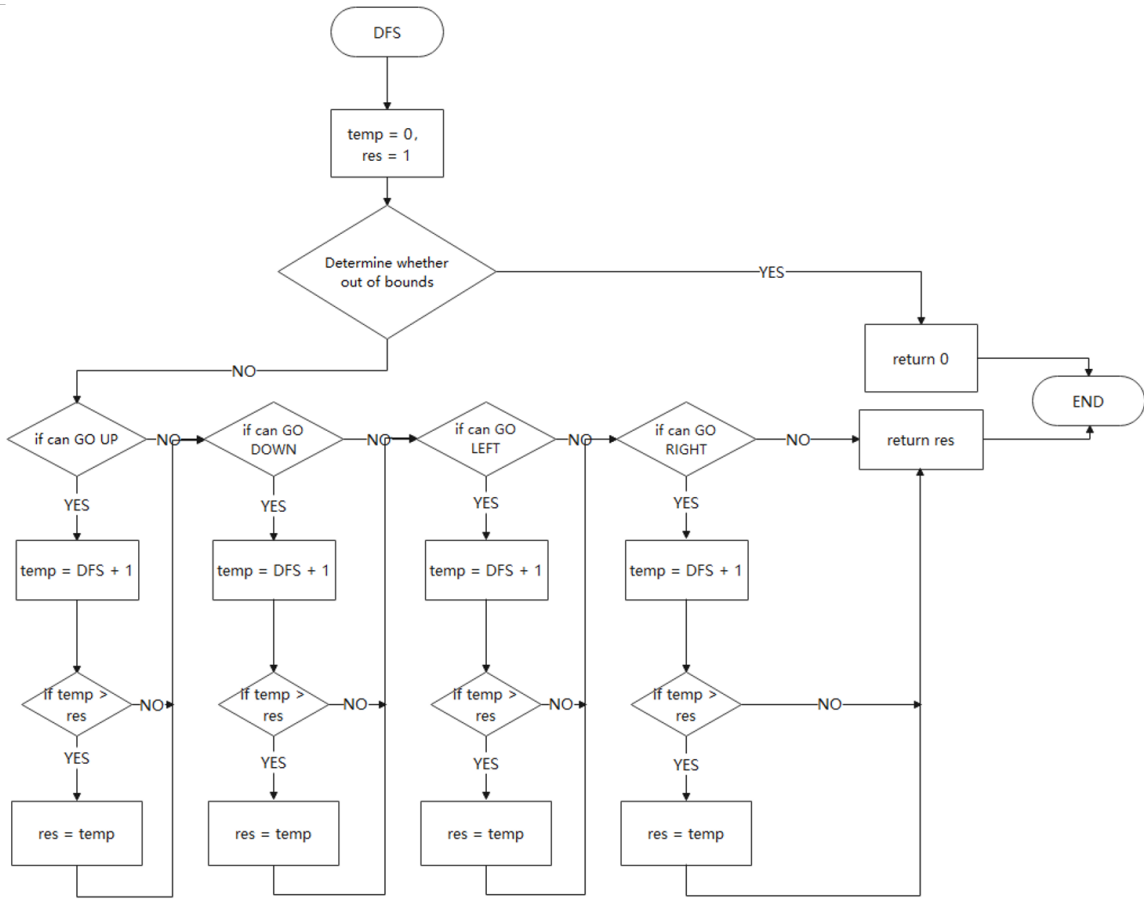
Lab 5 Report

> Student name: 王晨雨
>
> Strudent ID: 3200102324
>
> TA Check date: 2021/7/29
>
> Report submission date: 2021/7/29

1. Algorithm

This chart below shows the algorithm of the recursive program `DFS`



Main program use DFS to go through every point, looking for the longest path:

```
for(int i = 0; i < n; i++)
{
    for(int j = 0; j < m; j++)
    {
        int tempMax = DFS(a, i, j);
        if(tempMax > Max)
            Max = tempMax;
    }
}
```

2. Essential parts of code

For example, the following part of code shows `if(a[x][y] > a[x-1][y])`

```
                ; Go Up: if(a[x][y] > a[x-1][y])
NEXT1           LDR     R2, R5, #1      ; R2 = y, R1 = x
                LDR     R1, R5, #2
                ADD     R1, R1, #0
                BRz     NEXT2
                LD      R3, BASEN       ; R3 = x4000
                LDR     R4, R3, #1      ; R4 = m
                JSR     MULTIPLY
                ADD     R0, R2, R0
                ADD     R0, R0, #2      ; R0 = x*m+y+2
                ADD     R3, R3, R0
                LDR     R2, R3, #0      ; R2 = a[x*m+y+2]
                NOT     R4, R4
                ADD     R4, R4, #1
                ADD     R3, R3, R4
                LDR     R1, R3, #0      ; R1 = a[(x-1)*m+y+2]
                NOT     R1, R1
                ADD     R1, R1, #1
                ADD     R1, R2, R1
                BRnz    NEXT2

                LDR     R2, R5, #1      ; R2 = y, R1 = x-1
                LDR     R1, R5, #2
                ADD     R1, R1, #-1
                ADD     R6, R6, #-2     ; Push arguments
                STR     R1, R6, #1
                STR     R2, R6, #0
                JSR     DFS

                LDR     R3, R5, #-3     ; R3 = res, R4 = temp
                ADD     R4, R4, #1
                ADD     R0, R4, #0      ; R0 = R4
                NOT     R4, R4
                ADD     R4, R4, #1
                ADD     R4, R3, R4      ; R4 = res-temp
                BRzp    NEXT2
                ADD     R3, R0, #0      ; res = temp
                ADD     R4, R3, #0
                STR     R4, R5, #-3
```

The loop to find longest path:

```
                ; Find the maximum
                LD      R3, BASEN
                LDR     R1, R3, #0      ; R1 = n-1
OUTERLOOP       ADD     R1, R1, #-1
                BRn     TERMINATE
                LD      R3, BASEN
                LDR     R2, R3, #1      ; R2 = m-1

INNERLOOP       ADD     R2, R2, #-1
                BRn     OUTERLOOP
                ; Push arguments
```

```
12              ADD     R6, R6, #-2
13              STR     R1, R6, #1
14              STR     R2, R6, #0
15              ST      R0, SaveR0
16              ST      R1, SaveR11
17              ST      R2, SaveR22
18              JSR     DFS
19              LD      R0, SaveR0
20              LD      R1, SaveR11
21              LD      R2, SaveR22
22              ADD     R3, R4, #0      ; R3 = R4
23              NOT     R4, R4
24              ADD     R4, R4, #1
25              ADD     R4, R0, R4      ; R4 = Max-tempMax
26              BRzp    INNERLOOP
27              ADD     R0, R3, #0      ; Max = tempMax
28              BR      INNERLOOP
```

3. TA's questions

TA: Explain the procedure to write a recursive program.

Answer:

- Push arguments into stack
- Push control link stack
- Push return address into stack

- Push temorary variables
- (main body of subroutine)
- Pop the return address
- return

TA: How to judge the boundaries?

Answer:

- judge if x<0 or x>=n or y<0 or y>=m

```
1   if(x < 0 || x >= n || y < 0 || y >= m)
2       return 0
```