

Assignment Report 6

1. Solution

First, I set all the map points as -1 to show that there is no such point of height, which is very useful because the program will not need to judge whether it has crossed the boundaries.

```
for (i=0; i<MAX_NUM; i++) //set the height as -1 to make the boundaries
{
    for (j=0; j<MAX_NUM; j++)
    {
        Map[i][j] = -1;
    }
}
```

Then I read all the data (not from 0 to r-1 and 0 to c-1 but 1 to r and 1 to c because Map[0][j] and Map[i][0] are all boundaries).

Then I used memorized search to improve the program. If the current point is not 0 then it is already the max number, then just skip the procedure, or search the four directions to find the max numbers of the path by recursion.

```
if (Find[x][y] != 0) //if it is not 0, then the max result of this point has been found
    return Find[x][y];
for (i=0; i<4; i++)
{
    temp_x = x + dx[i];
    temp_y = y + dy[i];

    if (Map[temp_x][temp_y] != -1 && Map[temp_x][temp_y] < Map[x][y]) //find the max of the four directions
    {
        temp = SearchMaxPath(temp_x, temp_y)+1;
        if (Find[x][y] < temp) Find[x][y] = temp;
    }
}
```

Finally, the max number needs plus 1 because the last point had not been counted.

3. Source Code

```
#include <stdio.h>
#include <math.h>
#define MAX_NUM 21

int Map[MAX_NUM][MAX_NUM]; //data of the map
int Find[MAX_NUM][MAX_NUM]; //the result of memorized search

int dx[] = {-1, 0, 1, 0}; //four directions
```

```

int dy[] = {0, -1, 0, 1};

int SearchMaxPath(int x, int y)
{
    int i, temp;
    int tempx, tempy;
    if (Find[x][y] != 0) //if it is not 0, then the max result of this point has been found
        return Find[x][y];
    for (i=0; i<4; i++)
    {
        tempx = x + dx[i];
        tempy = y + dy[i];

        if (Map[tempx][tempy] != -1 && Map[tempx][tempy] < Map[x][y])//find the max of the
four directions
        {
            temp = SearchMaxPath(tempx, tempy)+1;
            if (Find[x][y] < temp) Find[x][y] = temp;
        }
    }

    return Find[x][y];
}

int main()
{
    int r, c;
    int i, j;
    int max, temp;
    scanf("%d %d", &r, &c);
    for (i=0; i<MAX_NUM; i++ ) //set the height as -1 to make the boundaries
    {
        for (j=0; j<MAX_NUM; j++)
        {
            Map[i][j] = -1;
        }
    }
    for (i=1; i<=r; i++) //read the data
    {
        for (j=1; j<=c; j++)
        {
            scanf("%d", &Map[i][j]);
            Find[i][j] = 0;
        }
    }
}

```

```

    }
}
max = 0;
for (i=1; i<=r; i++)
{
    for (j=1; j<=c; j++)
    {
        temp = SearchMaxPath(i, j);
        if (max < temp) max = temp;
    }
}
printf("%d", max + 1 ); //count the last point
return 0;
}

```

4. Snapshots (From Xcode)

```

5 5
1 2 3 4 5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9
25Program ended with exit code: 0

```