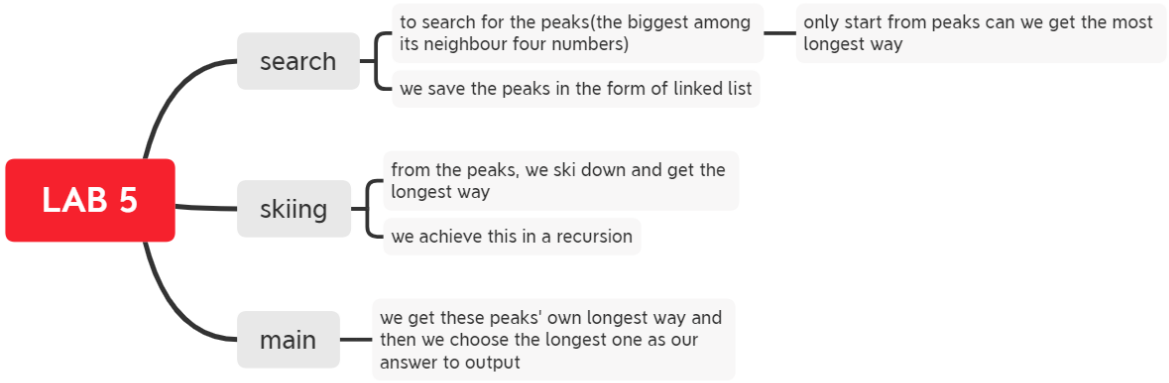
Lab 5 report



1. write in C

firstly I write in C to get a clear mind.

And I use this to show my mind in my report because it is more clear.

```c
#include<stdio.h>
#include<malloc.h>

#define length 5
#define height 5
int alt[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}};
//data struct of the peaks
struct peak {
    int i, j;
    struct peak *next;
} ;

int max(int a,int b)
{
    if(a>b) return a;
    return b;
}
//we get a linked list of the peaks from the function
struct peak* Searchforpeak(void)
{
    int i, j;
    struct peak *first, *now, *end;
    int *pos;
    int dfs = 0;

    first = (struct peak*)malloc(sizeof (struct peak));
    first->next = NULL;
    first->i = 0;
```

```
    first->j = 0;
    now = first;
    end = first;

    for(i = 0; i < 3;i++)
    {
        for(j = 0; j < 4; j++)
        {
            dfs = 0;//use dfs to decide whether it is peak.
            if(alt[i][j] < alt[i+1][j] && i+1 < height) dfs = 1;
            if(alt[i][j] < alt[i-1][j] && i-1 >= 0) dfs = 1;
            if(alt[i][j] < alt[i][j-1] && j-1 >= 0) dfs = 1;
            if(alt[i][j] < alt[i][j+1] && i+1 < length) dfs = 1;
            if(dfs == 0)
            {
                now = (struct peak*)malloc(sizeof (struct peak));
                now->i = i;
                now->j = j;
                now->next = NULL;
                end->next = now;
                end = now;
            }
        }
    }
    return first;
}
//
int skiing(int i, int j)
{
    int a[4] = {0};
    int maxium;

    if(alt[i][j] > alt[i+1][j] && i+1 < height) a[0] = skiing(i+1, j);
    if(alt[i][j] > alt[i-1][j] && i-1 >= 0) a[1] = skiing(i-1, j);
    if(alt[i][j] > alt[i][j-1] && j-1 >= 0) a[2] = skiing(i, j-1);
    if(alt[i][j] > alt[i][j+1] && j+1 < length) a[3] = skiing(i, j+1);

    maxium = max(max(a[0],a[1]),max(a[2],a[3]));//get the biggest one
    return maxium+1;
}

int main()
{
    int maxium, num, i, j;
    struct peak *first, *ptr;

    maxium = 1;
    first = Searchforpeak();
    ptr = first;
    while(ptr->next != NULL)
    {
        ptr = ptr->next;
        i = ptr->i;
        j = ptr->j;
        num = skiing(i, j);
        if(num > maxium) maxium = num;
    }
    printf("%d", maxium);
```

```
}
```

2. write in LC-3 assembly language

Then I translate it into LC-3 assembly language sentence by sentence and get the subroutines we want.

main function

```
main    JSR search
        AND R0, R0, #0;R0存maxium
        ADD R0, R0, #1
        LEA R1, peak
LOOP    LDR R2, R1, #1
        LDR R3, R1, #2
        ST  R0, SAVE_R0
        ST  R1, SAVE_R1
        ADD R6, R6, #-2
        STR R3, R6, #1
        STR R2, R6, #0
        JSR skiing  ;得到的极值放到R4当中
        LDR R4, R6, #0
        ADD R6, R6, #3
        LD  R0, SAVE_R0
        LD  R1, SAVE_R1
        ST  R4, SAVE_R4
        NOT R4, R4
        ADD R4, R4, #1
        ADD R2, R0, R4
        BRzp    LOOP
        LD  R4, SAVE_R4
        AND R0, R0, #0
        ADD R0, R0, R4
        LDR R1, R1, #0
        BRz exit
        BRnzp   LOOP
exit    out
        HALT
SAVE_R0 .BLKW #1
SAVE_R1 .BLKW #1
SAVE_R4 .BLKW #1
```

search function

1. initialization

   for all the numbers in the table:

2. check four times to compare with the neighbouring four numbers
3. if it is the biggest, add it to the linked list

and finally the skiing function

1. initialization

   for all the positions:

2. skiing from the four directions and get the longest way by recursion
3. and return to the peak we get the longest way

Because of the lack of space I didn't put all my codes there.