Practice 6 (2015/11/3)

Knight Tour on a Chessboard

1. Conditional Compilation

A. *#define, #ifdef, #ifndef, #else, #endif*

B. Example

☞ Version 1

```c
#include <stdio.h>
int main()  {
    int i, j, ret_no;

    printf("Please input two numbers:");
    ret_no = scanf("%d %d",&i, &j);
    printf("\nThe legal number of inputs are %d\n",ret_no);
    /* printf("\nThe legal number of inputs are %d\n",scanf("%d %d",&i,&j)); */
    return 1;
}
```

☞ Version 2

```c
#include <stdio.h>
int main()  {
    int i, j, ret_no;

    printf("Please input two numbers:");
    /*
    ret_no = scanf("%d %d",&i, &j);
    printf("\nThe legal number of inputs are %d\n",ret_no);
    */
    printf("\nThe legal number of inputs are %d\n",scanf("%d %d",&i,&j));
    return 1;
}
```

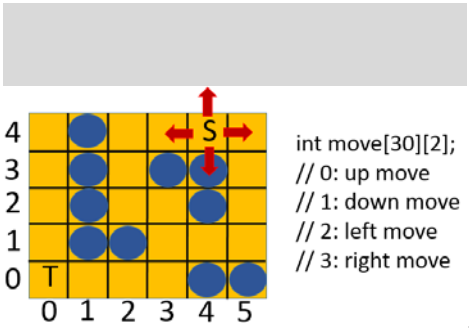
☞ Version 3

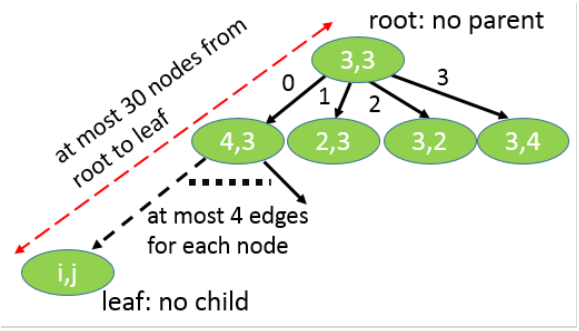
```c
#include <stdio.h>
#define _V1
int main()  {
    int i, j;

#ifdef _V1
    int ret_no;
#endif

    printf("Please input two numbers:");
#ifdef _V1
    ret_no = scanf("%d %d",&i, &j);
    printf("\nThe legal number of inputs are %d\n",ret_no);
#else
    printf("\nThe legal number of inputs are %d\n",scanf("%d %d",&i,&j));
#endif

    return 1;
}
```

2. Exhaustive Searching or Brute-Force

Brute-force method is commonly used by programming beginners and engineers. You can also call it exhaustive-searching method. Brute-force method tries to

explore every possible solution and reports the first solution it finds or the best solution after it scans all combinations in the solution space. Due to the significantly increasing computation power of computers, brute-force becomes a feasible way to solve a problem if it is affordable to scan all solution space by current computing power. With the simple techniques we have learned so far we also can solve some interesting problems with brute-force method. For instance, the problem of finding a shortest path from source to target on a chessboard also can be solved by brute-force.





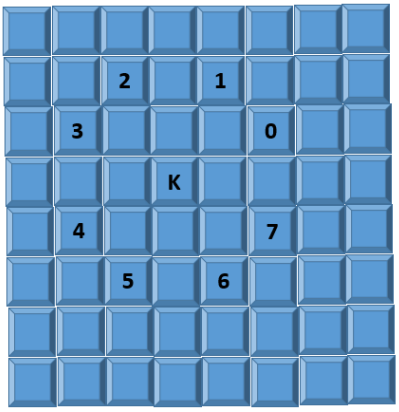
grid[5][6], tMove[4][2], path[30];
walk = 0; initialize path[] to -1;
while (walk >= 0) {
while (1) {
Try to move forwards by one step and keep move direction to path;
If reach target report path and exit program;
If one more step fails then exit this loop;
}
Get back to previous walk by one step
}

3. Problem

One of the more interesting puzzlers for chess buffs is the Knight's Tour problem, originally proposed by the mathematician Euler. The question is: Can the chess piece called the knight move around an empty chessboard and touch each of the 64 squares once and only once? The knight make L-shape moves (over two in one direction and then over one in a perpendicular direction). Thus from a square in the middle of an empty chessboard, the knight can make eight different moves (numbered 0 through 7) as shown in the following. Write a program to see how many steps your knight can make. You can define 8 types of moves for your knight and define movType for each type of move (in the following figure). Then the location after one move is:

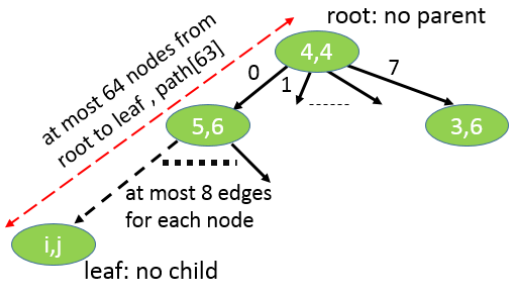
curRow += verMove[movType];

curCol += horMove[movType];



horMove[0] = 2; horMove[1] = 1;
horMove[2] = -1; horMove[3] = -2;
horMove[4] = -2; horMove[5] = -1;
horMove[6] = 1; horMove[7] = 2;

verMove[0] = 1; verMove[1] = 2;
verMove[2] = 2; verMove[3] = 1;
verMove[4] = -1; verMove[5] = -2;
verMove[6] = -2; verMove[7] = -1;



	1	2	3	4	5	6	7	8
1	7	42	27	12	5	10	17	22
2	28	13	6	9	16	21	4	19
3	41	8	15	26	11	18	23	36
4	14	29	0	39	24	35	20	3
5	0	40	25	0	0	0	37	34
6	30	0	0	0	38	33	2	0
7	0	0	0	32	0	0	0	0
8	0	31	0	0	0	0	0	1

Total moves = 41

	1	2	3	4	5	6	7	8
1	27	6	25	20	9	4	15	12
2	24	21	8	5	14	11	18	3
3	7	26	23	10	19	2	13	16
4	22	0	0	1	0	17	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Total moves = 26

Main features:

(a) You can change the initial location to find a longest path you can move. You have to print the moves you have made on the screen like the figures above (due tonight)

(b) Use brute-force method to move your Knight an empty chessboard and touch each of the 64 squares once and only once. (due in 11/7)