

1-Climbing Stair

Name	ID
يوسف ياسر محمد رضوان	201900999
محمد رضا راضي عبدالواحد	201900669
جرجس ميلاد لويس نسيم	201900241
حسن هاني حسن مهدي	201900268
احمد فتحي انور محمد	20160036
عبد الرحمن صلاح مجاهد ذكي	20210511

1- Recursive Algorithm:

The pseudocode:

```
1- Function RecursiveclimbStairs (x )
2- // X is The input that the user will enter
3- {
4- IF x <= 1
5- return 1
6 – ELSE
7 – return RecursiveclimbStairs(x – 1 ) + RecursiveclimbStairs(x – 2)
8 – // Then Call This function in The Main()
9- Function Main
10- Declear n
11-Print (“Enter the number of stairs ”)
12-Read n
13- call Function RecursiveclimbStairs(n)
```

The time complexity :

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \\ &= 2^2T(n-2) + 2O(1) + O(1) \\ &= 2^2(2T(n-3) + O(1)) + 2O(1) + O(1) \\ &= 2^3T(n-3) + 2^2O(1) + 2O(1) + O(1) = \end{aligned}$$

$$2^k T(n-k) + 2^{k-1} O(1) + \dots + 2 O(1) + O(1)$$

where $k = n - 1$.

$$T(n) = 2^{n-1} O(1) + 2^{n-2} O(1) + \dots + 2 O(1) + O(1)$$

$$= O(2^n)$$

The Output:

The screenshot shows a C++ IDE with the following code in `main.c`:

```

1 #include <stdio.h>
2
3 int climbStairsRecursive(int n) {
4     if (n <= 1) {
5         return 1;
6     } else {
7         return climbStairsRecursive(n-1) + climbStairsRecursive(n-2);
8     }
9 }
10
11 int main() {
12     int n;

```

The output window shows the execution results:

```

Select "C:\Users\lap\Desktop\Algo\Recursive Algorithm\bin\Debug\Recursive Algorithm.exe"
Enter the number of stairs: 4
Number of distinct ways to climb the stairs: 5
Process returned 0 (0x0)   execution time : 1.549 s
Press any key to continue.

```

The taskbar at the bottom indicates the system is running Windows 10, with the date 5/4/2021 and time 3:19 PM.

2- The Non Recursive Algorithm:

The pseudocode:

```
1-Function NON RecursiveclimbStairs (x)
2- // X is The input that the user will enter
3- IF x <= 1
4- return 1
5- ELSE
6- a:= 1 , b := 1
7- SET c
8- for l := 2 To x DO
9- c := a + b
10- a := b
11- b := c
12- return b
13- END Function
14- // Call this function in the Main()
15- Function main()
16- DECLARE n
17-PRINT "Enter the number of stairs: "
18-READ n
19-PRINT "Number of distinct ways to climb the stairs: ",
climbStairsNonRecursive(n)
20- return 0
21-END Function
```

The time complexity :

1- a single comparison operation ($n \leq 1$) The Return Statement is 1

So the time = $O(1)$

2- Enter the Else Statement :

3 - initializes three integer variables Take the Time = $O(1)$

4 – Then Enter the For Loop from 2 To N So The Time = $O(n - 2)$

5-Then The Function return Variable b So The Time = $O(1)$

6 – SO The TOTAL TIME COMPLEXITY = $O(n)$

The Output:

```
#include <stdio.h>
int climbStairsNonRecursive(int n) {
    if (n <= 1) {
        return 1;
    } else {
        int a = 1, b = 1, c;
        for (int i = 2; i <= n; i++) {
            c = a + b;
            a = b;
            b = c;
        }
        return b;
    }
}
```

Select "C:\Users\lap\Desktop\Algo\ Non Recursive Algorithm\bin\Debug\ Non Recursive Algorithm.exe"

Enter the number of stairs: 4
Number of distinct ways to climb the stairs: 5
Process returned 0 (0x0) execution time : 2.973 s
Press any key to continue.

The Comparison:

	Recursive Algorithm:	Non Recursive Algorithm:
Time Complexity	$O(2^n)$	$O(n)$
Space Complexity	$O(n)$	$O(1)$

From the previous table, we see that the use of the Non Recursive Algorithm is better than the use of the Recursive Algorithm in terms of time complexity, as the time in the Non Recursive Algorithm is $O(2^n)$

and therefore less than the Recursive Algorithm and also better than it in terms of Space complexity where the Space complexity in the Non Recursive Algorithm is $O(1)$ So The Non Recursive Algorithm is better.