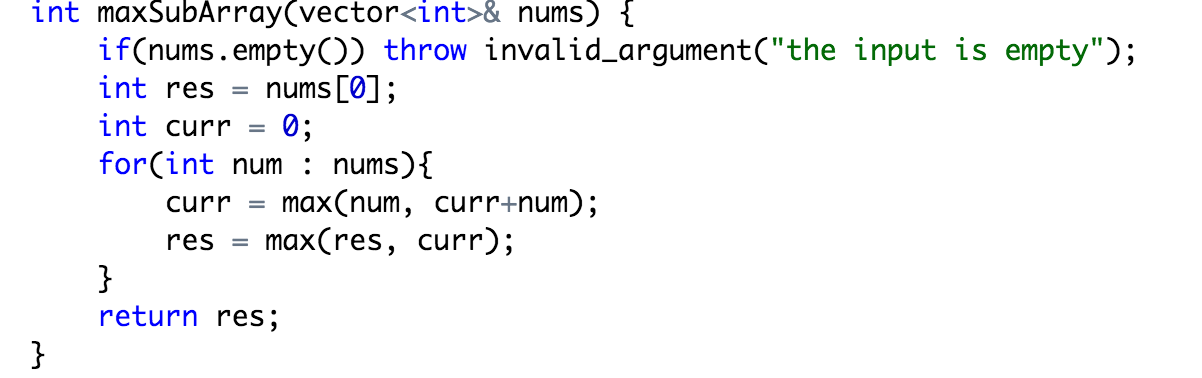
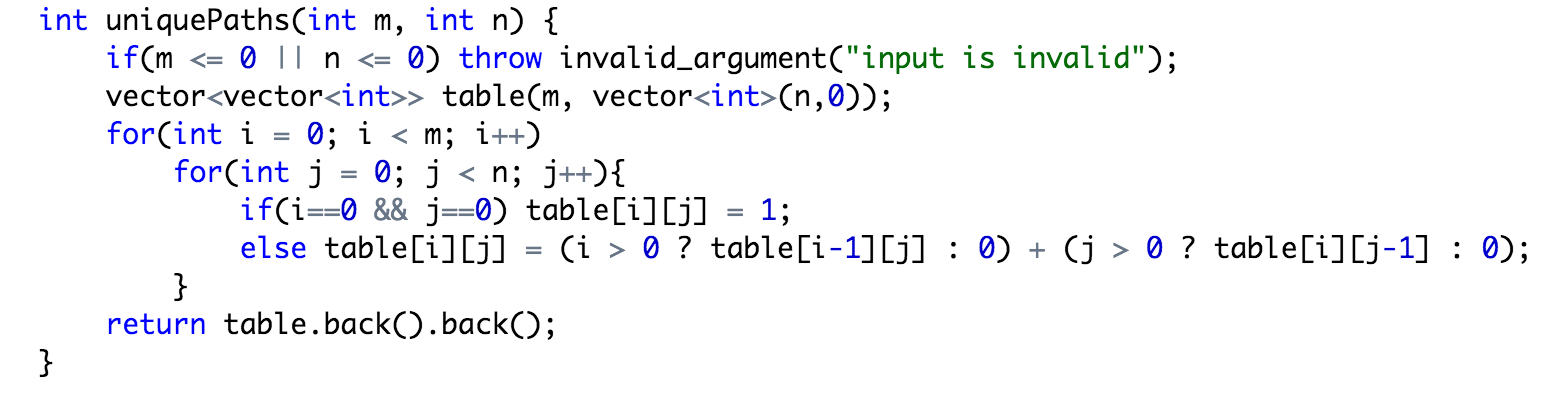
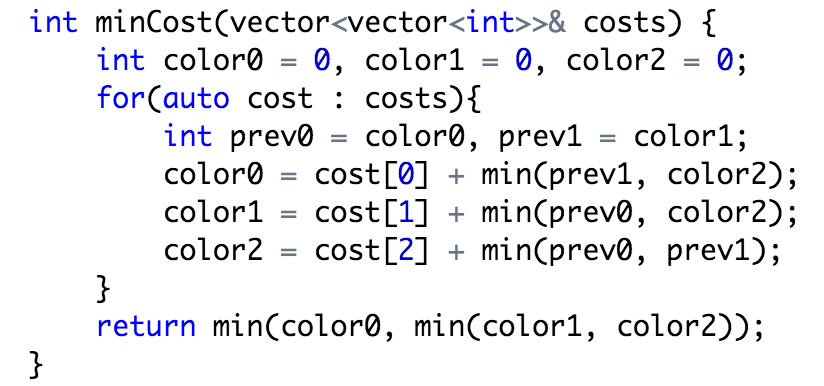
**1. Maximum Subarray**

****

**2. Unique Paths**

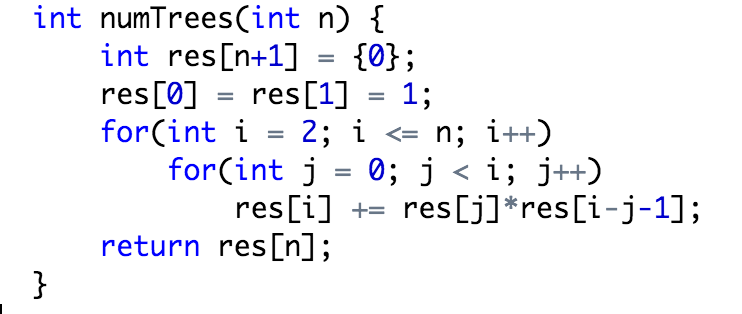


**3. Paint House**



**4. Unique Binary Search Trees**

if(n <= 0) throw invalid\_argument(“invalid input”);

****

**5.Fibonacci Number**

Given an integer n, write a function to return the Nth number in the Fibonacci series.

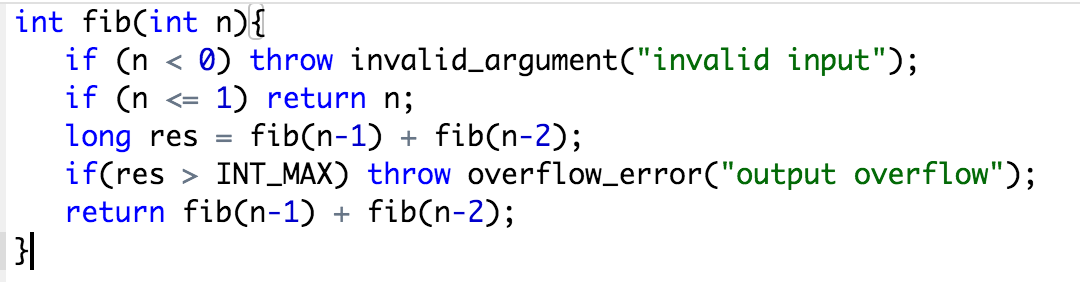
Corner Case:

1. input invalid 负数

2. n = 0, 1, 2 fib(0) = 0; fib(1) = 1; fib(i) = fib(i-1) + fib(i-2);

3.overflow

**Recursion Version**

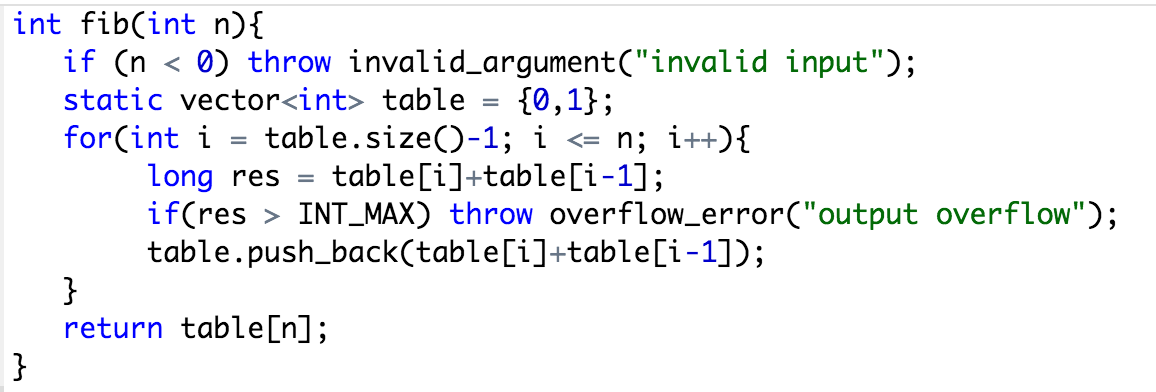


Time: 2^n

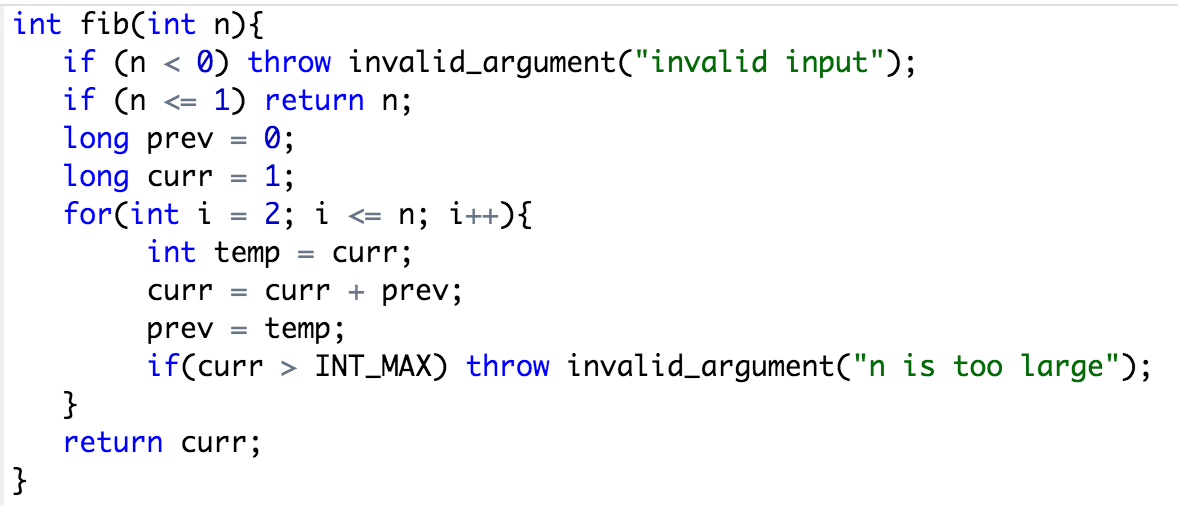
Space: O(n) if we consider the function call stack size, otherwise O(1).

Improve Time O(n): use to hash map 存中间值。

**DP Version 1**



**DP Version 2**



**Follow Up Fibonacci return小于M的最后N个数**

Queue with size N

**Follow Up怎么判定一个数是fibonacci数**

负数不是, 算出第一个大于等于该数的fibonacci，如果是等于就是，大于就不是

**Follow Up如果那个数N很大，如何处理**

Cut down the time complexity to O(log(N)) by computing the multiplication of N matrices.



