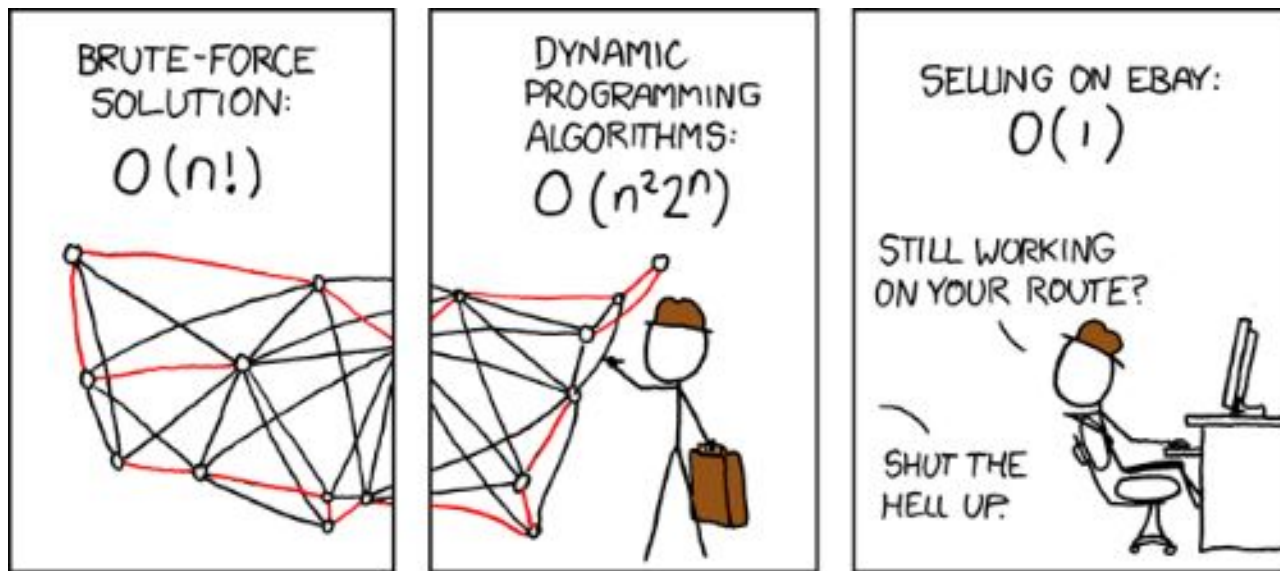




Дивизион Е. День 3  
Динамическое программирование


Лектор: Дмитрий Руденко




The Traveling Salesman...

# Fibonacci Recursion and Dynamic Programming

## Числа Фибоначчи

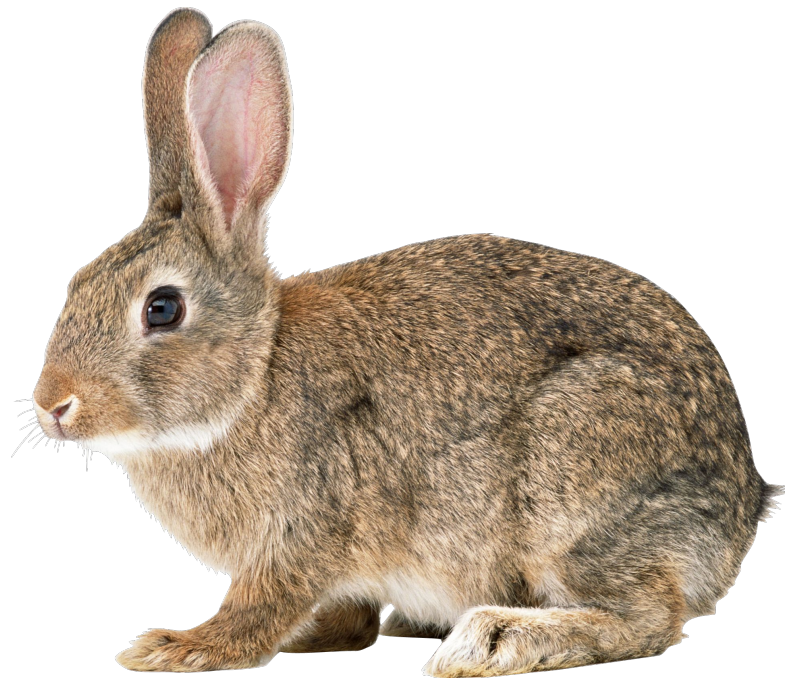


```
int fib(int n) {  
    if (n ≤ 1) {  
        return 1;  
    } else {  
        return fib(n - 1) + fib(n - 2);  
    }  
}
```



```
vector<int> dp(n + 1);  
  
dp[0] = dp[1] = 1;  
  
for (int i = 2; i ≤ n; i++) {  
    dp[i] = dp[i - 1] + dp[i - 2];  
}
```

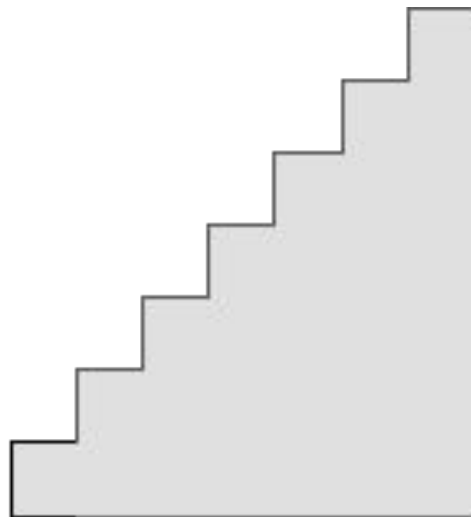
# Задача о зайчике



# Задача о количестве путей

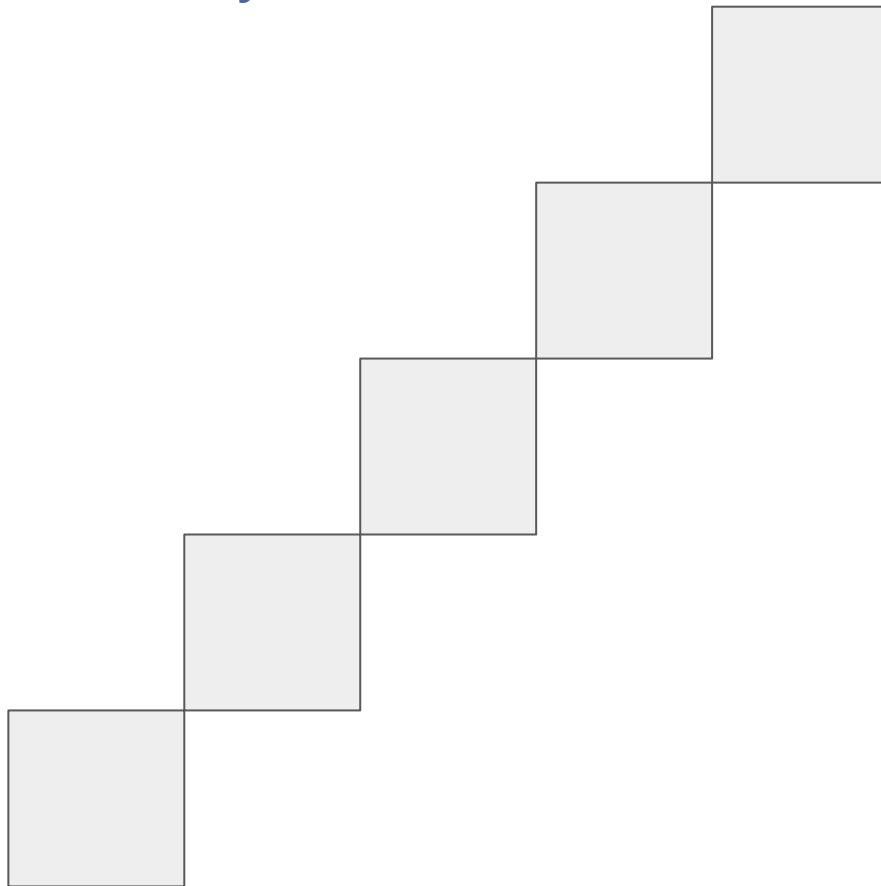
## Условие

Есть  $N$  ступеней и заяц, который хочет добраться наверх. Он может совершать прыжки высотой в 1 и 2 ступени. Помогите определить зайцу сколько у него существует способов достичь цели.



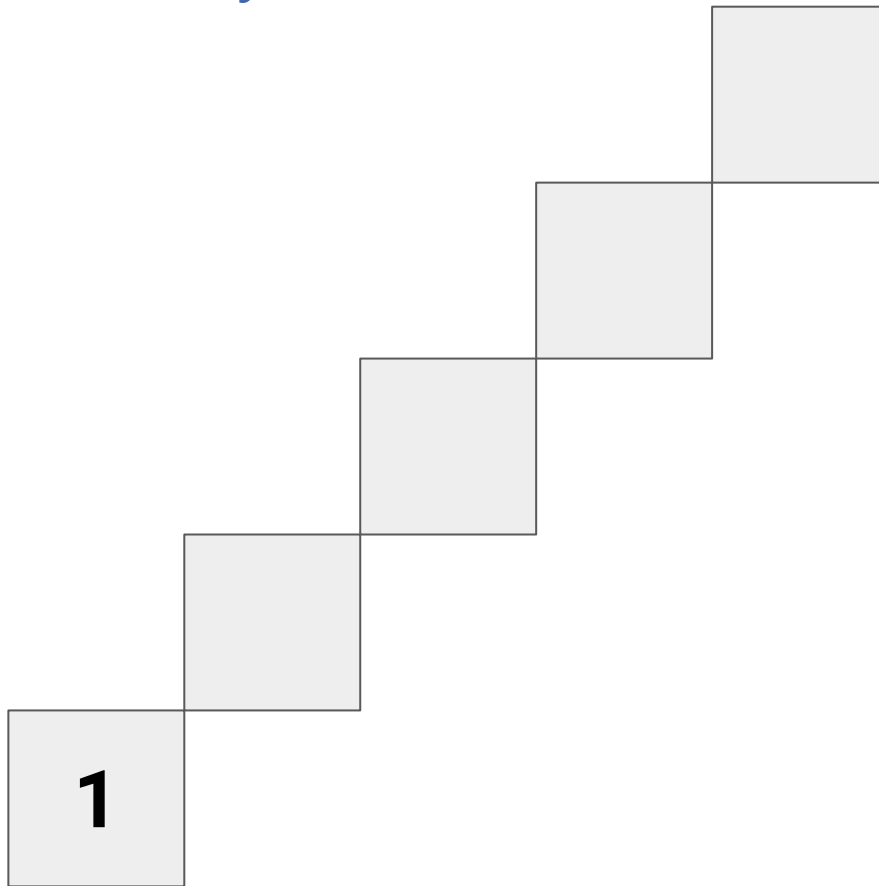
## Задача о количестве путей

Пол →



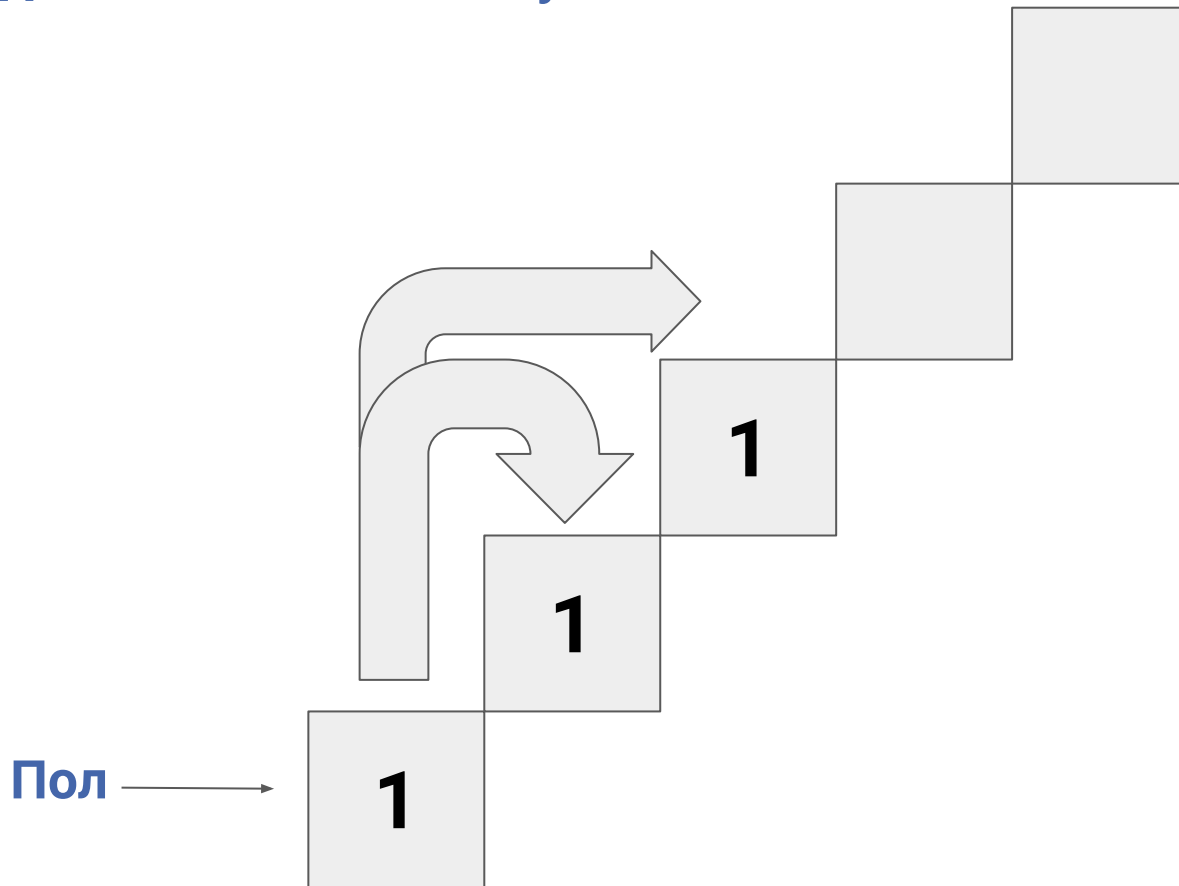
## Задача о количестве путей

Пол →

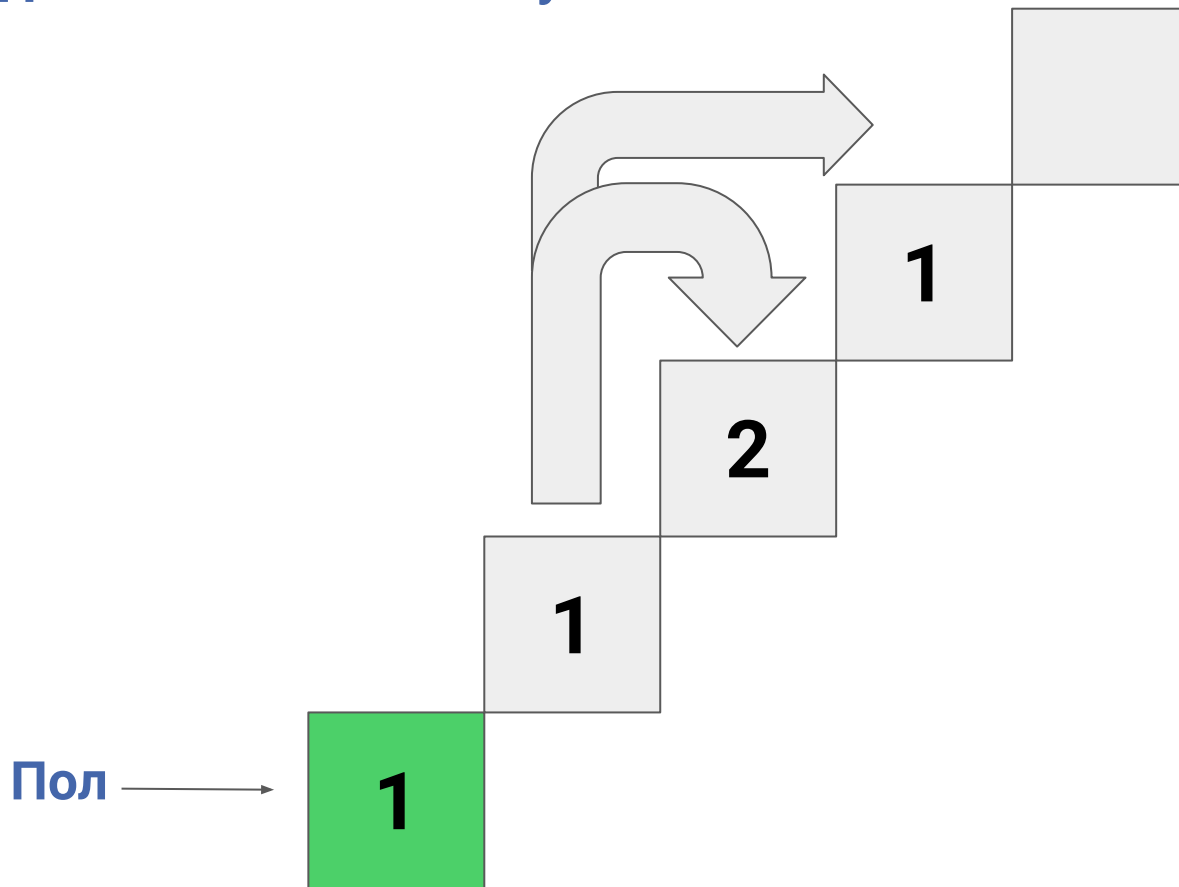




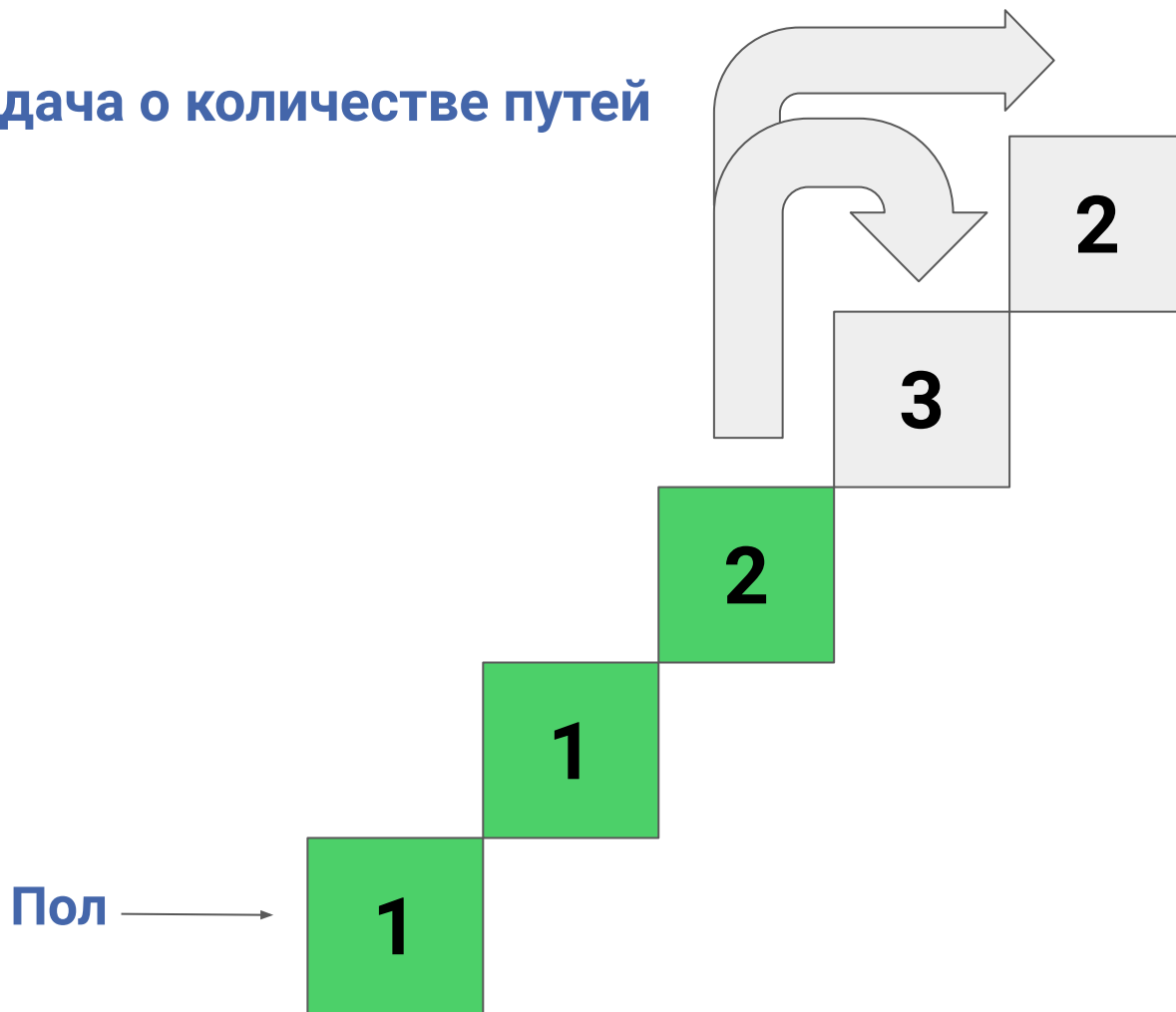
## Задача о количестве путей



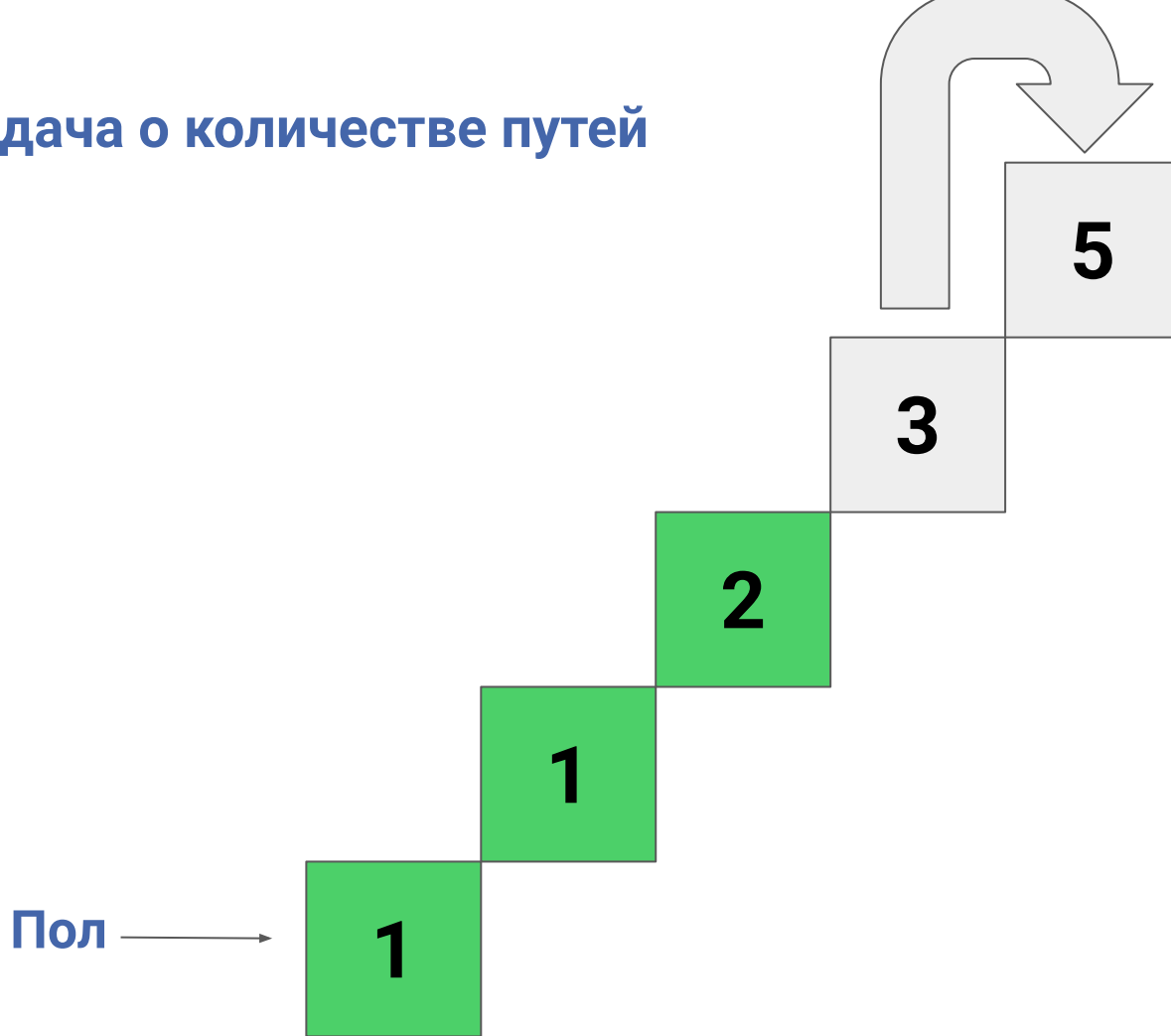
## Задача о количестве путей



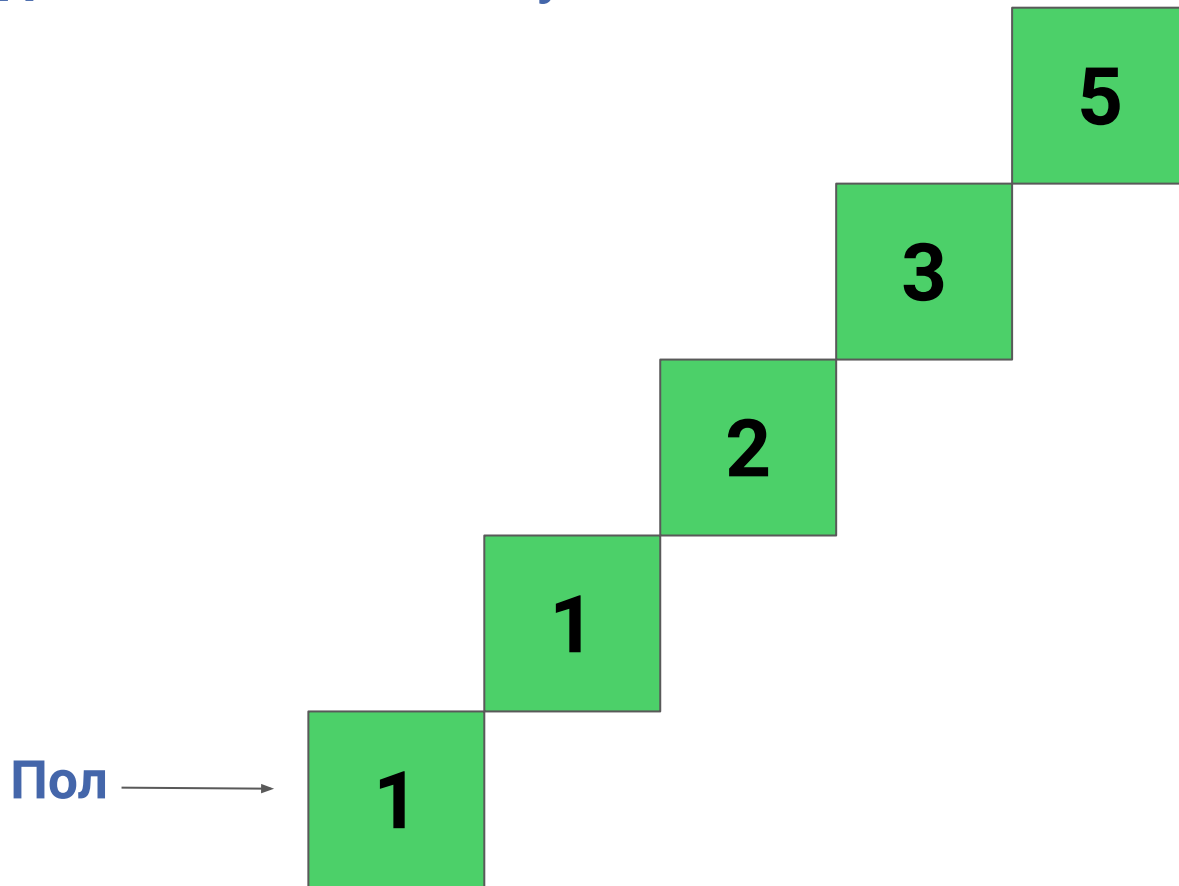
## Задача о количестве путей



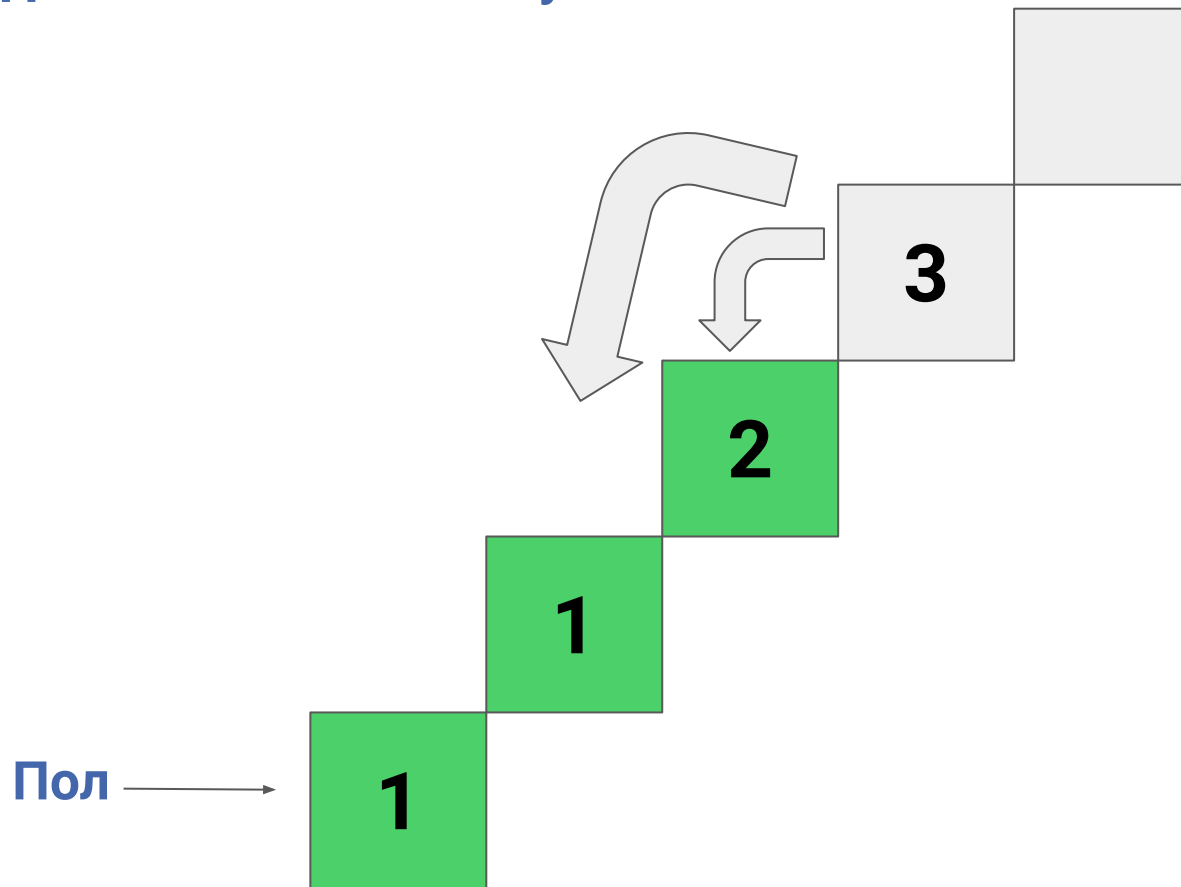
## Задача о количестве путей



## Задача о количестве путей



## Задача о количестве путей



# Задача о битовой последовательности

## Условие

Посчитать количество различных битовых последовательностей длины  $N$ , у которых нет двух подряд идущих нулей.



# Задача о битовой последовательности

## База динамики

$dp[0] = 1$  // ""

$dp[1] = 2$  // '1' и '0'





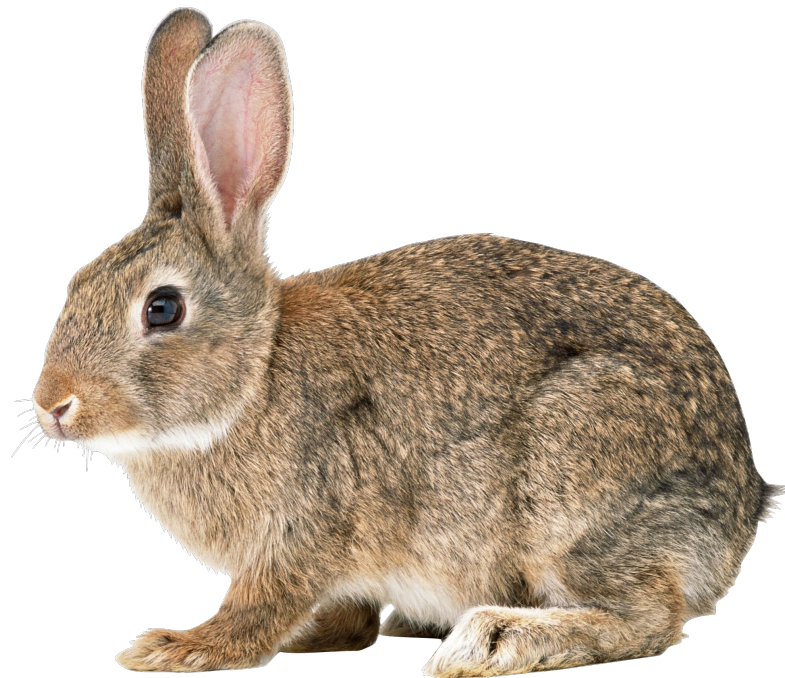
# Задача о битовой последовательности

## Переход

- 1)  $dp[i]$  - ?
- 2) Пусть  $i = 5$ :
- 3)  $????1 \rightarrow dp[5] = dp[4] + \dots$
- 4)  $????0$
- 5)  $???10 \rightarrow dp[5] = dp[3] + \dots$
- 6)  $dp[5] = dp[4] + dp[3]$
- 7)  $dp[i] = dp[i - 1] + dp[i - 2]$



# Задача о зайчике



## Задача о количестве путей в таблице

Старт






Финиш

## Задача о количестве путей в таблице

Старт



1			



Финиш

## Задача о количестве путей в таблице

Старт

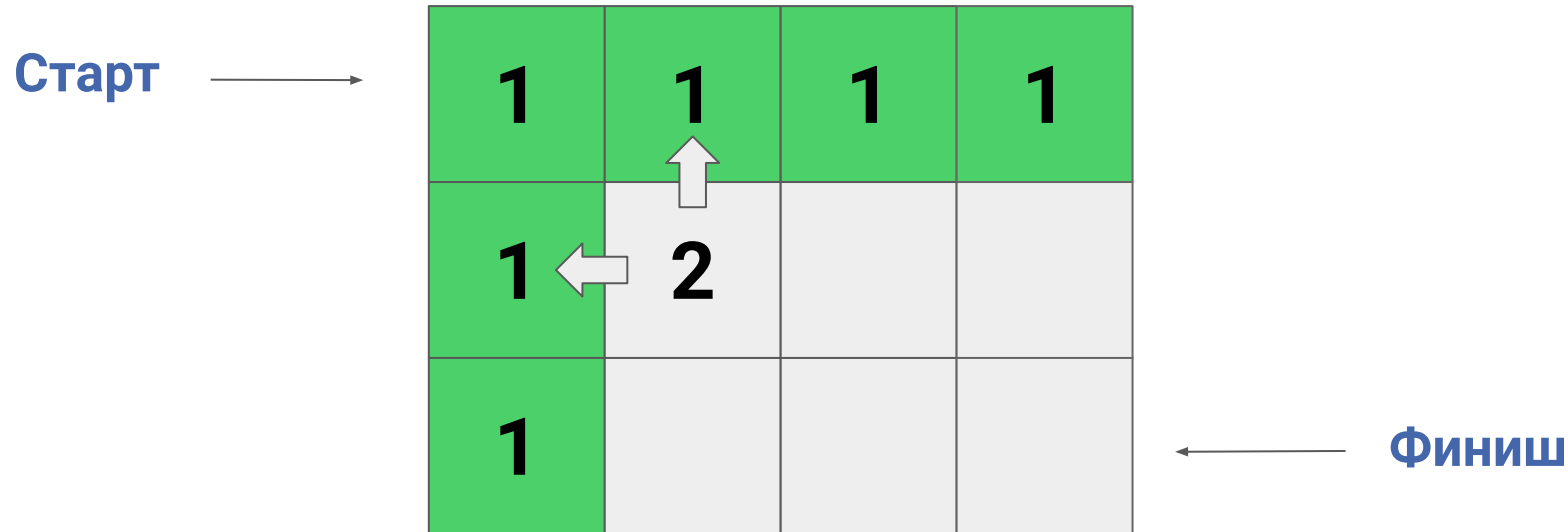


1	1	1	1
1			
1			



Финиш

## Задача о количестве путей в таблице



## Задача о количестве путей в таблице

Старт



1	1	1	1
1	2	3	
1			



Финиш

## Задача о количестве путей в таблице

Старт



1	1	1	1
1	2	3	4
1			



Финиш



## Задача о количестве путей в таблице

Старт



1	1	1	1
1	2	3	4
1	3		



Финиш

## Задача о количестве путей в таблице

Старт



1	1	1	1
1	2	3	4
1	3	6	



Финиш

## Задача о количестве путей в таблице

Старт



1	1	1	1
1	2	3	4
1	3	6	10



Финиш

## Задача о количестве путей в таблице

Старт



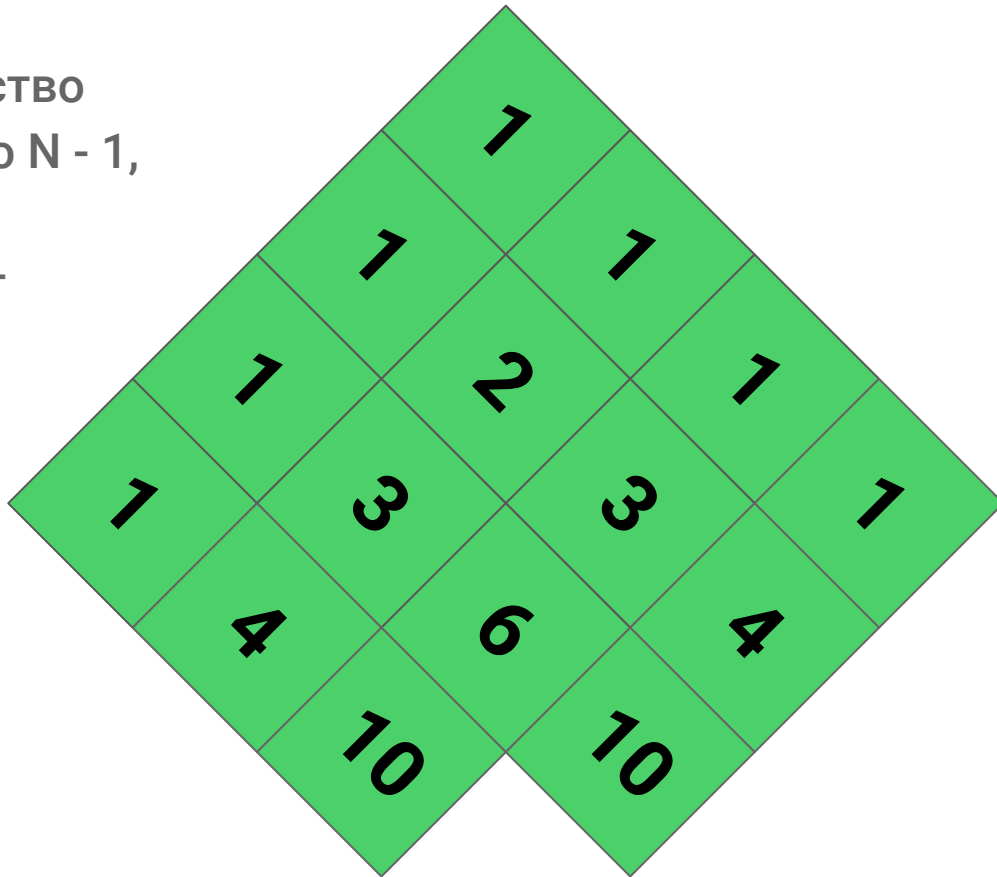
1	1	1	1
1	2	3	4
1	3	6	10



Финиш

## Задача о количестве путей в таблице

Также ответ это количество сочетаний из  $N + M - 2$  по  $N - 1$ ,  
или из  $N + M - 2$  по  $M - 1$   
( $N - 1$  шаг вниз,  $M - 1$  шаг  
вправо - все  $N + M - 2$ )



# Задача о количестве путей в таблице

---



```
vector<vector<int>>> dp(n, vector<int>(m));

for (int i = 0; i < n; ++i) {
    dp[i][0] = 1;
}

for (int j = 0; j < m; ++j) {
    dp[0][j] = 1;
}

for (int i = 1; i < n; ++i) {
    for (int j = 1; j < m; ++j) {
        dp[i][j] = dp[i - 1][j] + dp[i][j - 1];
    }
}
```

## Минимизация стоимости

Старт



5	2	0	2
3	1	3	1
8	4	6	2



Финиш

## Минимизация стоимости

Старт



5	2	0	2
3	1	3	1
8	4	6	2



Финиш



## Минимизация стоимости

Старт



5	7	0	2
3	1	3	1
8	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	2
3	1	3	1
8	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
3	1	3	1
8	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	1	3	1
8	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	1	3	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	1	3	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	3	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	3	1
16	4	6	2



Финиш



## Минимизация стоимости

Старт



5	7	7	9
8	8	3	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	1
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	4	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	12	6	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	12	16	2



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	12	16	2



Финиш



## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	12	16	12



Финиш

## Минимизация стоимости

Старт



5	7	7	9
8	8	10	10
16	12	16	12



Финиш

## Минимизация стоимости

Старт



5	2	0	2
3	1	3	1
8	4	6	2



Финиш

# Минимизация стоимости



```
vector<vector<int>> dp(n, vector<int>(m));  
vector<vector<int>> a(n, vector<int>(m));  
  
dp[0][0] = a[0][0];  
  
for (int i = 1; i < n; ++i) {  
    dp[i][0] = a[i][0] + dp[i - 1][0];  
}  
  
for (int j = 1; j < m; ++j) {  
    dp[0][j] = a[0][j] + dp[0][j - 1];  
}  
  
for (int i = 1; i < n; ++i) {  
    for (int j = 1; j < m; ++j) {  
        dp[i][j] = a[i][j] + min(dp[i - 1][j], dp[i][j - 1]);  
    }  
}
```