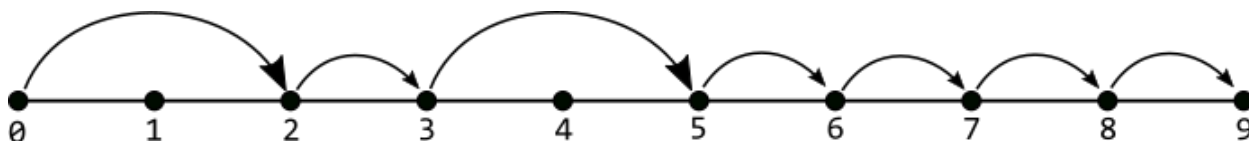


Семинар #6: Динамическое программирование. Классные задачи.

Задача о кузнечике:

Кузнечик находится в начале координата (точке $x = 0$) и может прыгать вперёд на 1 шаг или на 2 шага. Найти количество разных путей которыми кузнечик может добраться из точки 0 до точки $x = n$. Пример одного из путей на рисунке:



Решение:

Пусть $f(k)$ – это количество всевозможных путей от точки $x = 0$ до точки $x = k$. Предположим, что мы знаем все $f(k)$ до $k = n - 1$. Как найти $f(n)$?

В точку n кузнечик может попасть из точки $n - 1$ или из точки $n - 2$. Число возможных путей до точки n равно сумме числа путей до точки $n - 1$ и числа путей до точки $n - 2$:

$$f(n) = f(n - 1) + f(n - 2)$$

Решение на языке C:

```
#include <stdio.h>
#define MAX 100

int main()
{
    int f[MAX] = {};
    int n;
    scanf("%d", &n);
    f[0] = 1;
    f[1] = 1;

    for (int i = 2; i <= n; ++i)
    {
        f[i] = f[i-1] + f[i-2];
    }
    printf("%d\n", f[n]);
}
```

Задачи:

- **СуперКузнечик:** Предположим, что кузнечик может прыгать на любое число шагов. Найдите количество разных путей которыми кузнечик может добраться из точки $x = 0$ до точки $x = n$.
- **Кузнечик на плоскости:** Кузнечик находится на плоскости в точке $(0, 0)$ и может прыгать на 1 шаг вправо или на 1 шаг вверх. Найдите количество разных путей которыми кузнечик может добраться из точки $(0, 0)$ до точки (n, m) .

ВХОД	ВЫХОД
2 2	6
6 8	3003
10 10	184756

Задача о черепашке:

Черепашка находится в верхнем левом углу прямоугольной таблицы. В каждой ячейке таблицы лежит определенное число кочанов капусты. Черепашка может двигаться либо вниз либо вправо по таблице на 1 шаг. Найдите такой путь, при котором черепашка съест наибольшее число кочанов капусты.

0	→ 1	→ 2	→ 4	3	6	5	1
1	5	0	↓ 9	→ 9	1	2	2
8	7	2	0	↓ 9	→ 8	→ 7	4
20	10	1	4	3	1	↓ 8	→ 1

На вход задаче поступает файл с размером таблицы и самой таблицей. Нужно напечатать путь в виде строки из символов R (Right) и D (Down) и количество кочанов капусты на этом пути. Например, для таблицы на изображении программа должна выдать:

RRRDRDRRDR 58

Подмассив максимальной суммы:

Напишите программу, которая принимает на вход массив и находит непрерывный подмассив, имеющий наибольшую сумму. Например, для массива:

{5, -4, 6, -9, 7, 4, -9, 22, 4, 87, -73, 45, 3, 84, 5, -43, -38, -3, 74, -5}

подмассив максимальной суммы будет такой:

{7, 4, -9, 22, 4, 87, -73, 45, 3, 84, 5}

Подсказка:

$f(n)$ – в этой задаче – это сумма лучшего подмассива, который заканчивается в точке n . Также потребуется хранить $start(n)$ – индекс первого элемента подмассива, который заканчивается в точке n .