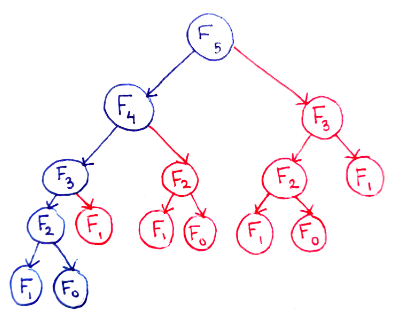
**Билет №8**

**Динамическое программирование. Основные понятия, 5 шагов для решения задачи методом ДП на примере задачи о черепашке.**

**Динамическое программирование. Основные понятия:**

**Динамическое программирование** — метод решения задачи путём её разбиения на несколько одинаковых подзадач, рекуррентно связанных между собой.

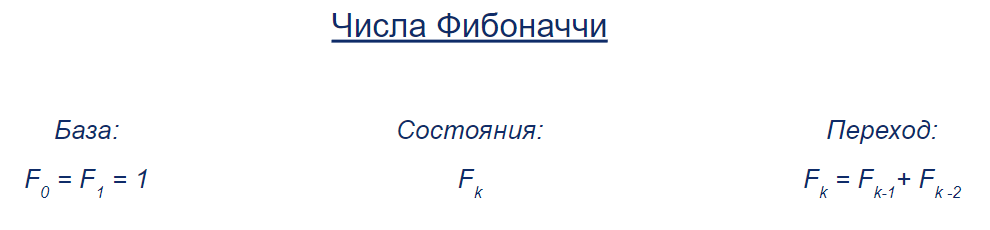
Самым простым примером будут числа Фибоначчи — чтобы вычислить некоторое число в этой последовательности, нам нужно сперва вычислить третье число, сложив первые два, затем четвёртое таким же образом на основе второго и третьего, и так далее.



**База** — известный ответ (или ответы) для задачи при фиксированных начальных условиях.

**Состояния** — подзадачи, на которые мы разбиваем изначальную (основную) задачу.

**Переход** — связь (обычно формула) для получения ответа для одного состояния через другие состояния.



**Определим основные шаги, которые необходимо проверить и продумать, чтобы решить задачу методом динамического программирования:**

1. Что хранится в dp[]? Продумать нужно как тип (Число, строка, пара чисел), так и что из себя представляет это значение (Количество способов, минимальное значение)
2. База dp? Результаты задач, которые мы можем использовать без пересчета. Как правило, эти ответы очевидны и используются в самом начале вычисления ответа
3. Формула пересчета? Она может быть не одна, однако нужно четко определить для каждой задачи, по какой формуле можно вычислить ответ на текущую задачу
4. Порядок пересчета? Необходимо пересчитывать в таком порядке, чтобы во время вычисления мы знали верные ответы для всех тех, которые мы используем в формуле
5. Где брать ответ?

**Задача:** на квадратной доске расставлены целые неотрицательные числа. Черепашка, находящаяся в левом верхнем углу, мечтает попасть в правый нижний. При этом она может переползать только в клетку справа или снизу и хочет, чтобы сумма всех чисел, оказавшихся у нее на пути, была бы максимальной. Определить эту сумму.

**Шаги для нашей задачи:**

1. Что хранится в dp[]? Тут нам нужно понять, что мы храним в dp, в задаче о черепашке обычно храним собранные монеты, значит нужно кол-во, а значит int/long
2. База dp? Результаты задач, которые мы можем использовать без пересчета. Как правило, эти ответы очевидны и используются в самом начале вычисления ответа. В задаче про черепашку это стартовая клетка. Первый ряд и первый столбец тоже, так как там, очевидно, что в ряду черепашка может двигаться только вниз, а в строке, что только вправо
3. Формула пересчета? Она может быть не одна, однако нужно четко определить для каждой задачи, по какой формуле можно вычислить ответ на текущую задачу. В задаче о черепашке о нахождении макс суммы формула для пересчёта будет едина для всех клеток, кроме тех, что описаны в пункте выше: dp[i][j] = max(dp[i-1][j], dp[i][j-1])+a[i][j]
4. Порядок пересчета? Необходимо пересчитывать в таком порядке, чтобы во время вычисления мы знали верные ответы для всех тех, которые мы используем в формуле. В задаче на черепашку мы используем динамику назад, так как используем вычисленные значения, а значит i = 2 to n, j = 2 to m, если нумерация с 1

Короче динамика назад, когда мы используем ранее вычисленные значения для i-го элемента, а динамика вперед, когда мы будущие вычисляем на основе i-го элемента(нихрена непонятно)

Ниже пример, когда нумерация идёт с 0:

for (int i = 1; i < n; i++) {

      for (int j = 1; j < m; j++) {

                arr[i][j] = Math.max(arr[i - 1][j] + gr[i][j], arr[i][j - 1] + gr[i][j]);

            } }

1. Где брать ответ? тут нужно понять в какой ячейке будет ответ, в задаче про черепашку он будет в dp[n][m] (если нумерация с 1)

**Решение:**

Обозначим доску как матрицу А. Пусть нам известен “максимальный” путь для всех клеток, кроме правой нижней - функция F(X, Y).

Все нужные маршруты проходят через одну из клеток, смежных с этим углом (их всего две). Максимальный же маршрут проходит через ту клетку из двух, для которой значение функции F больше.

Получается, что F(X, Y) = max(F(X-1; Y); F(X; Y-1)) + A[X; Y]

https://pastebin.com/wtAFinkJ

**Асимптотика:** O(n\*m).