



Дивизион Е. День 3
Динамическое программирование

Лектор: Дмитрий Руденко

Взрывоопасность

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) $dp[i] = dp[i - 1] + dp[i - 2]$



Взрывоопасность - 2

Переход

- 1) $dp[i]$ - ?
- 2) Пусть $i = 5$:
- 3) $????1 \rightarrow dp[5] = dp[4] + \dots$
- 4) $????2 \rightarrow dp[5] = dp[4] + \dots$
- 5) $????0$
- 6) $???10 \rightarrow dp[5] = dp[3] + \dots$
- 7) $???20 \rightarrow dp[5] = dp[3] + \dots$
- 8) $dp[5] = (dp[4] + dp[3]) * 2$
- 9) $dp[i] = (dp[i - 1] + dp[i - 2]) * 2$
- 10) База: $dp[0] = 1, dp[1] = 3$

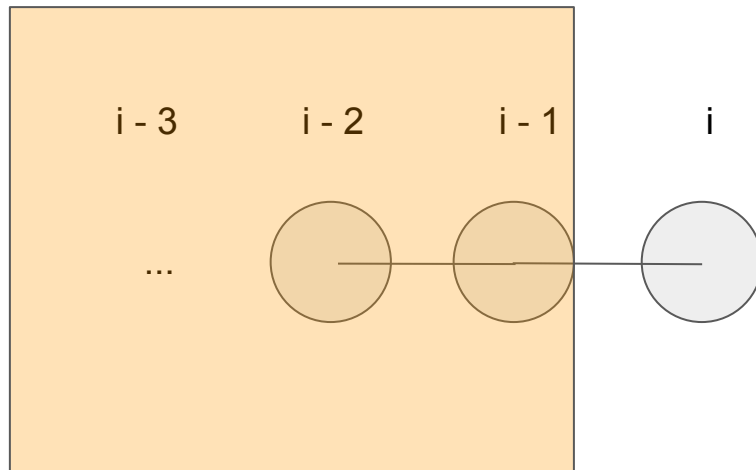
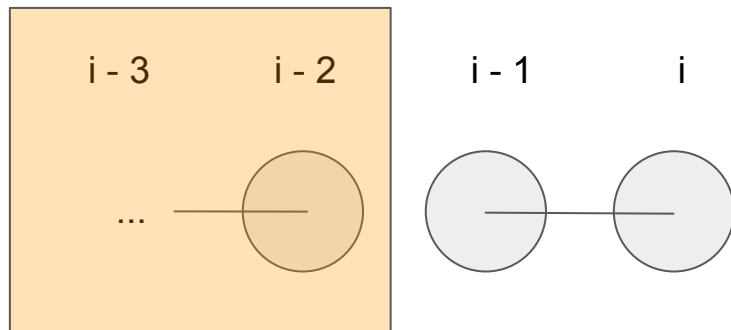


Гвоздики

- 1) Отсортируем гвозди по координатам
- 2) $dp[i]$ - минимальная суммарная длина если связать гвозди до i -го включительно
(нумерация с 0)
- 3) $dp[1] = a[1] - a[0]$
- 4) $dp[2] = dp[1] + a[2] - a[0]$

Гвоздики

- 1) $a[i] - a[i - 1] + dp[i - 2]$
- 2) $a[i] - a[i - 1] + dp[i - 1]$
- 3) Нужен минимум
- 4) $dp[i] = a[i] - a[i - 1] + \min(dp[i - 2], dp[i - 1])$



Банкомат

- 1) $dp[i]$ - минимальное количество купюр, чтобы выдать i денег
- 2) база динамики - $dp[0] = 0$, $dp[i] = INF$, $i \neq 0$
- 3) пусть у нас есть 3 купюры номиналом a , b и c
- 4) давайте посмотрим сколько купюр надо для $dp[i - a]$, $dp[i - b]$ и $dp[i - c]$ денег
- 5) $dp[i - b] \rightarrow 3$, $x + y + z = i - b$, $x + y + z + b = i$, 4 купюры
- 6) выберем минимальное значение и к этим купюрам добавим соответствующую
- 7) 8; 1, 3, 5
- 8) $dp[0] = 0$, $dp[1] = dp[1 - 1] + 1 = 1$, $dp[2] = dp[2 - 1] + 1 = 2$, $dp[3] = \min(dp[3 - 1], dp[3 - 3]) + 1$
- 9) $prev[2] = 1$, $prev[3] = 3$

Банкомат

- 1) Как выдать ответ?
- 2) Пусть `prev[i]` - последняя купюра в оптимальном решении для суммы `i` (`dp[i]`)
- 3) тогда если `dp[sum] != INF`, в цикле:
- 4) выводим `prev[sum]`
- 5) вычитаем из `sum` значение `prev[sum]`
- 6) повторяем, пока `sum != 0`

Банкомат

1) Пример:

2) 5 купюр: 1 3 7 12 32, сумма: 40

$dp[40] = \min(\dots, dp[40 - 32], \dots) + 1$

$dp[40] = 3, prev[40] = 32 // 1\ 7\ 32$

$40 - 32 \rightarrow 8$

$dp[8] = 2, prev[8] = 7 // 1\ 7$

$8 - 7 = 1$

$dp[1] = 1, prev[1] = 1$

$1 - 1 = 0$

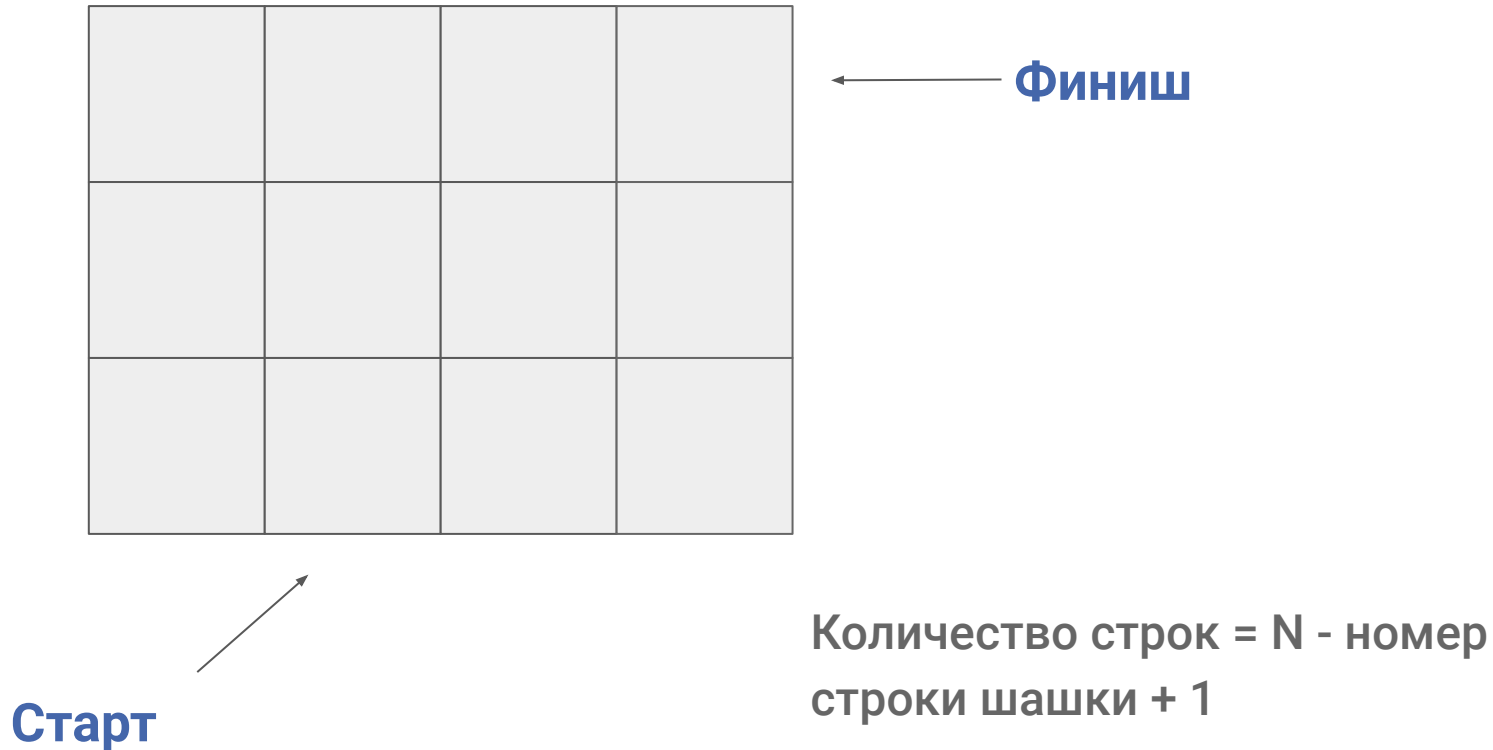
Банкомат

```
vector<int> dp(sum + 1, INF);
vector<int> prev(sum);
dp[0] = 0;
for (int i = 1; i ≤ sum; i++) {
    for (int j = 0; j < n; j++) {
        if (i - a[j] ≥ 0 && dp[i - a[j]] ≠ INF) {
            if (dp[i - a[j]] + 1 ≤ dp[i]) {
                dp[i] = dp[i - a[j]] + 1;
                prev[i] = a[j];
            }
        }
    }
}
```

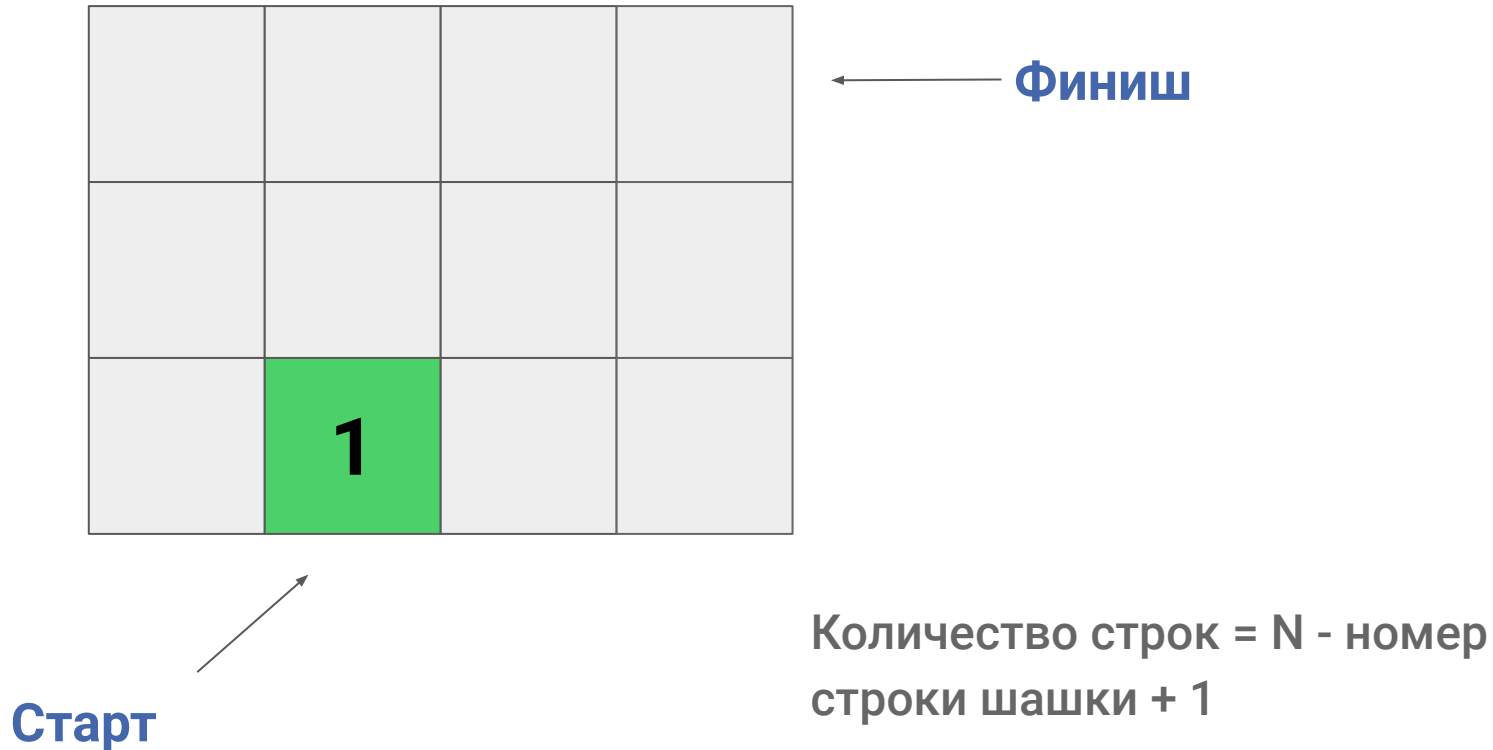
Банкомат

```
if (dp[sum] == INF) {  
    cout << -1;  
} else {  
    int curr_sum = sum;  
    cout << dp[sum] << "\n";  
    while (curr_sum != 0) {  
        cout << prev[curr_sum] << " ";  
        curr_sum -= prev[curr_sum];  
    }  
}
```

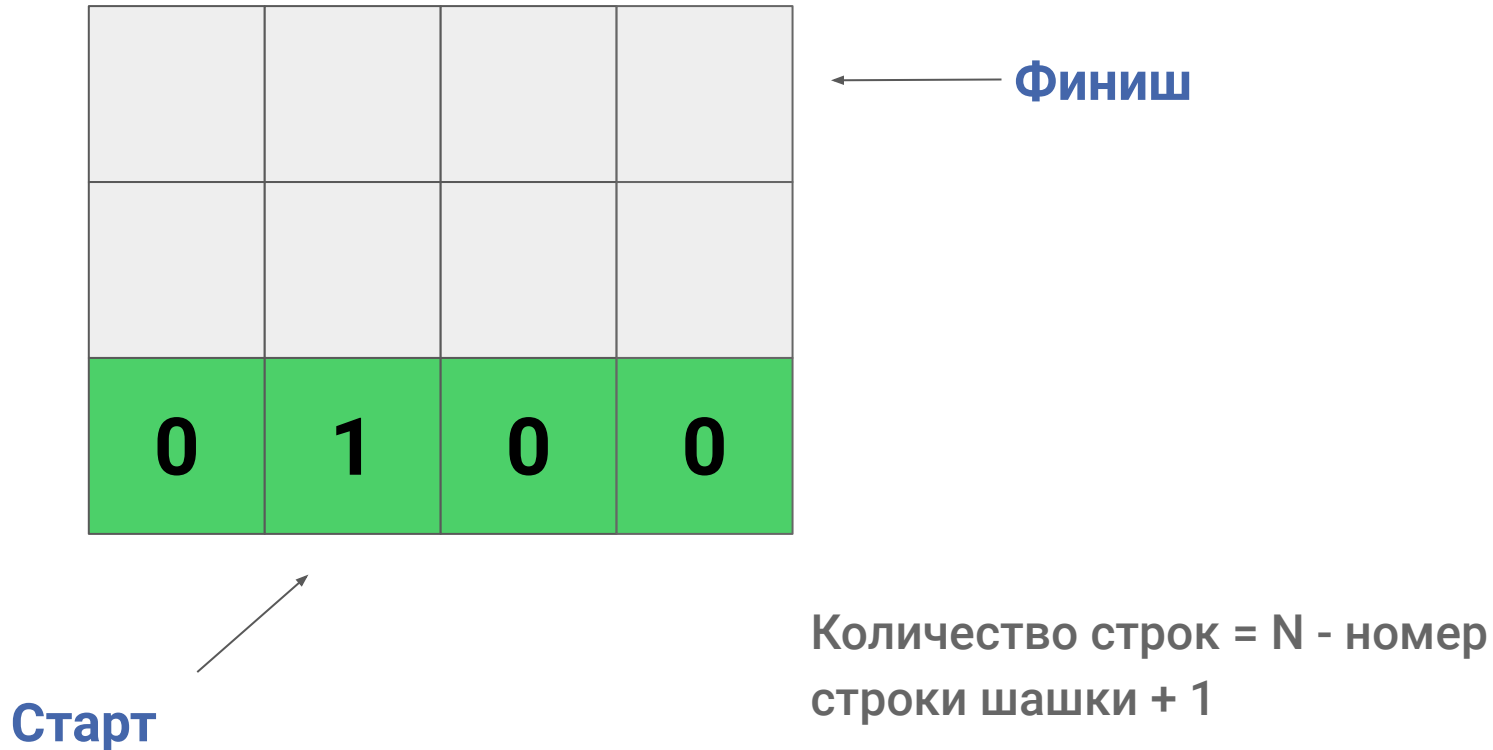
Задача про шашку



Задача про шашку



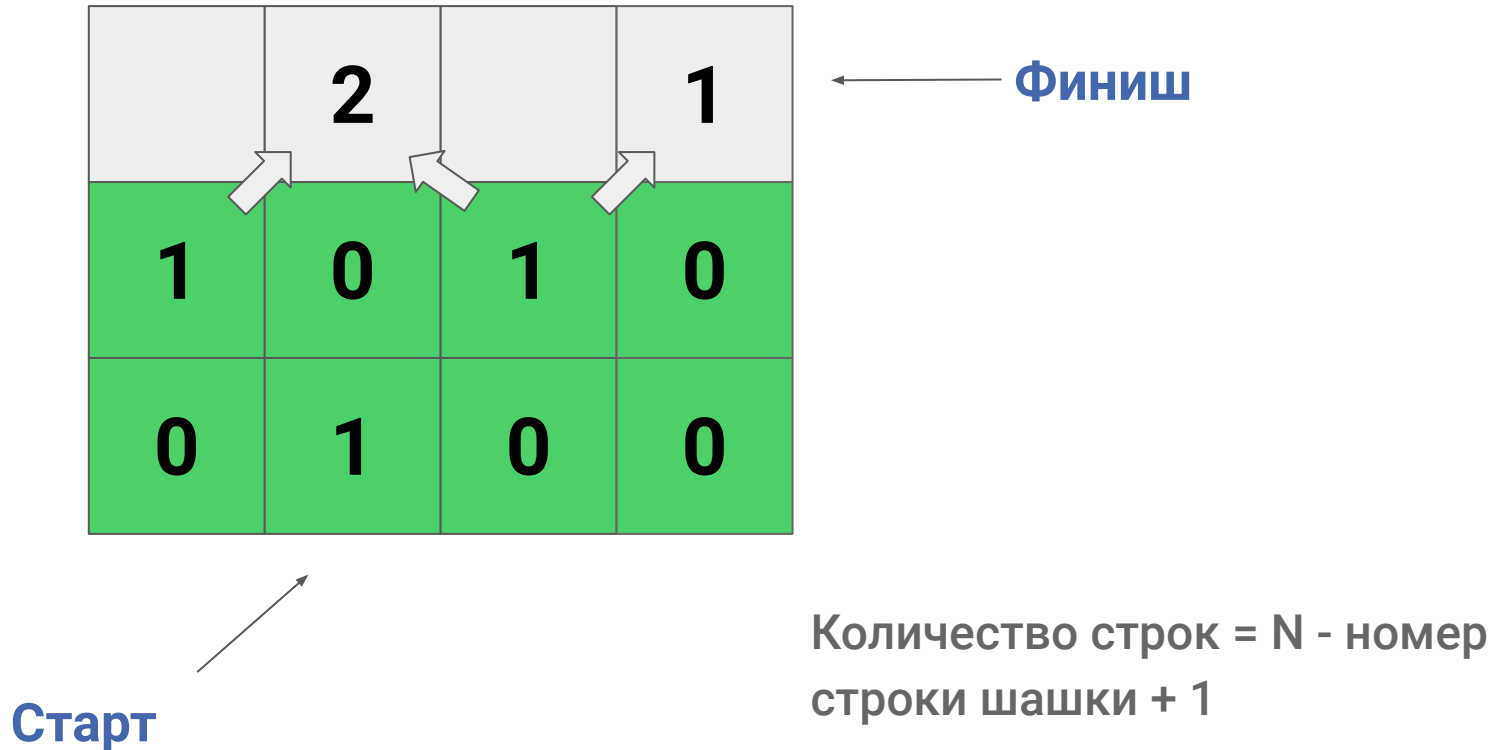
Задача про шашку



Задача про шашку



Задача про шашку



Задача про шашку

0	2	0	1
1	0	1	0
0	1	0	0

← **Финиш**

Старт ↗

Количество строк = N - номер
строки шашки + 1

Задача про шашку

0	2	0	1
1	0	1	0
0	1	0	0

← Финиш

↗
Старт

Ответ: сумма элементов
верхней строки

Задача про шашку

```
int n = 8;
int col, row;
cin >> col >> row;
col--;
vector<vector<int>> dp(n - row + 1, vector<int>(n, 0));

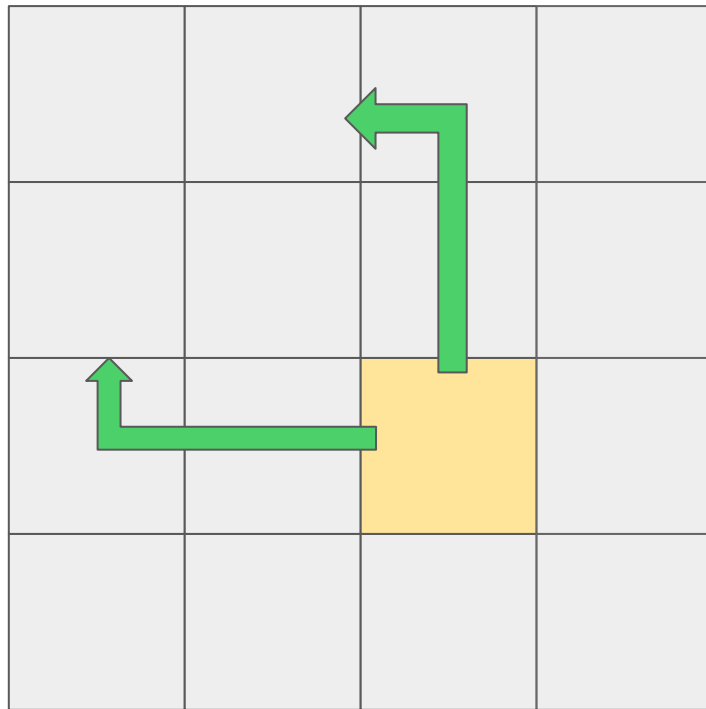
dp[n - row][col] = 1;

for (int i = n - row; i > 0; --i) {
    for (int j = 0; j < n; j++) {
        if (j > 0) {
            dp[i - 1][j - 1] += dp[i][j];
        }
        if (j < n - 1) {
            dp[i - 1][j + 1] += dp[i][j];
        }
    }
}
```

Задача про коня

$$dp[i][j] = dp[i - 1][j - 2] + dp[i - 2][j - 1]$$

Старт



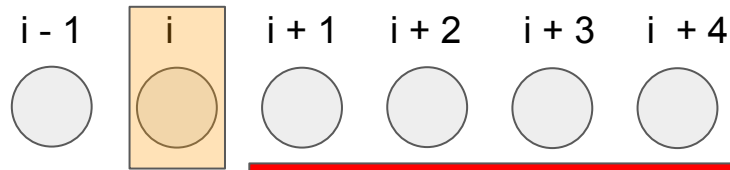
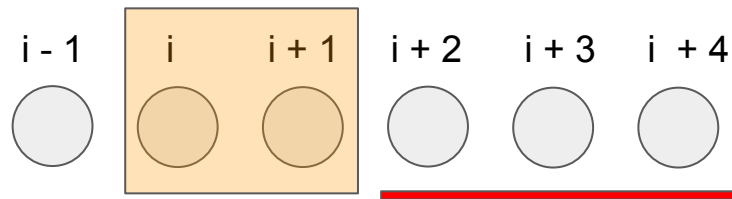
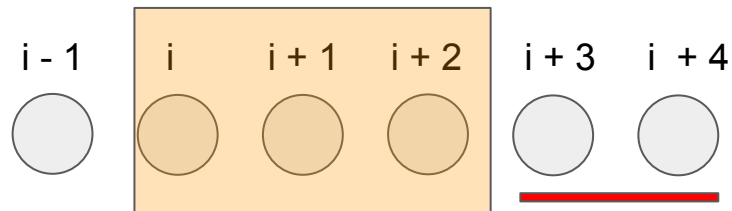
Финиш

Задача про билеты

- 1) $dp[i]$ - минимальное время за которое будут обслужены покупатели от последнего до i -го (то есть суффикс на очереди)
- 2) База:
 $dp[n - 1] = a[n - 1]$, потому что последний покупатель покупает только себе
 $dp[n - 2] = \min(b[n - 2], a[n - 2] + dp[n - 1])$, либо на двоих, либо каждый себе
 $dp[n - 3] = \min(c[n - 3], b[n - 3] + dp[n - 1], a[n - 3] + dp[n - 2])$

Задача про билеты

$dp[i] = \min(\begin{aligned} &c[i] + dp[i + 3], \\ &b[i] + dp[i + 2], \\ &a[i] + dp[i + 1] \end{aligned})$



Задача про билеты



```
vector<int> dp(n);  
dp[n - 1] = a[n - 1];  
if (n > 1) {  
    dp[n - 2] = min(a[n - 2] + dp[n - 1], b[n - 2]);  
}  
if (n > 2) {  
    dp[n - 3] = min(min(a[n - 3] + dp[n - 2], b[n - 3] + dp[n - 1]), c[n - 3]);  
}  
for (int i = n - 4; i ≥ 0; i--) {  
    dp[i] = min(min(a[i] + dp[i + 1], b[i] + dp[i + 2]), c[i] + dp[i + 3]);  
}
```