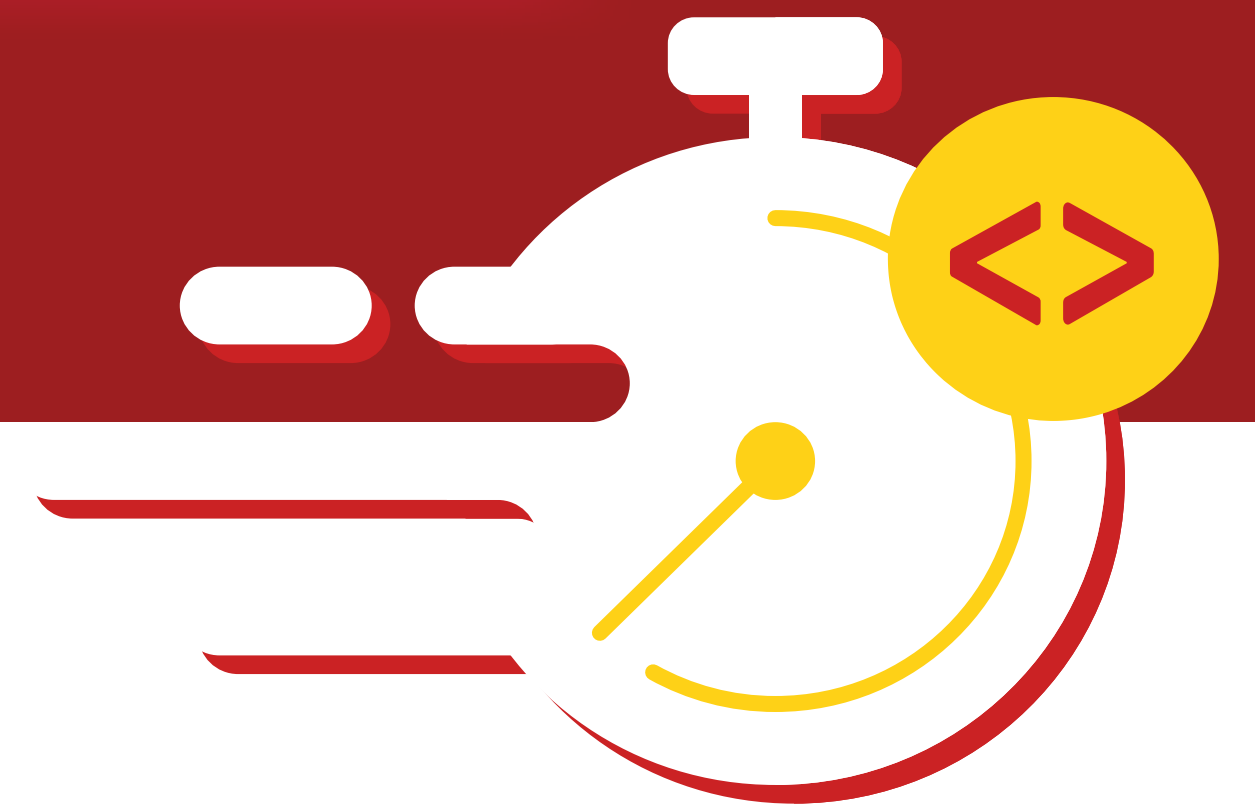
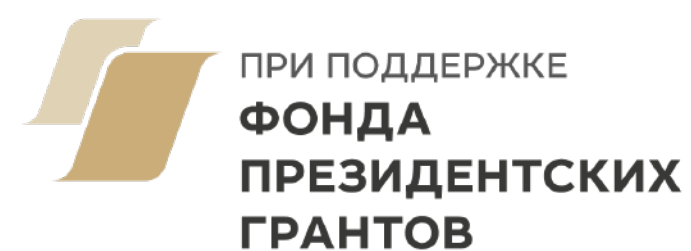


# Задача о рюкзаке

## Урок 3.3



# Задача о рюкзаке





# Постановка задачи\_

$n$



$w_1$



$w_2$



$w_3$

...



$w_n$



$W$

# Пример\_

6



2



3



4



4



6



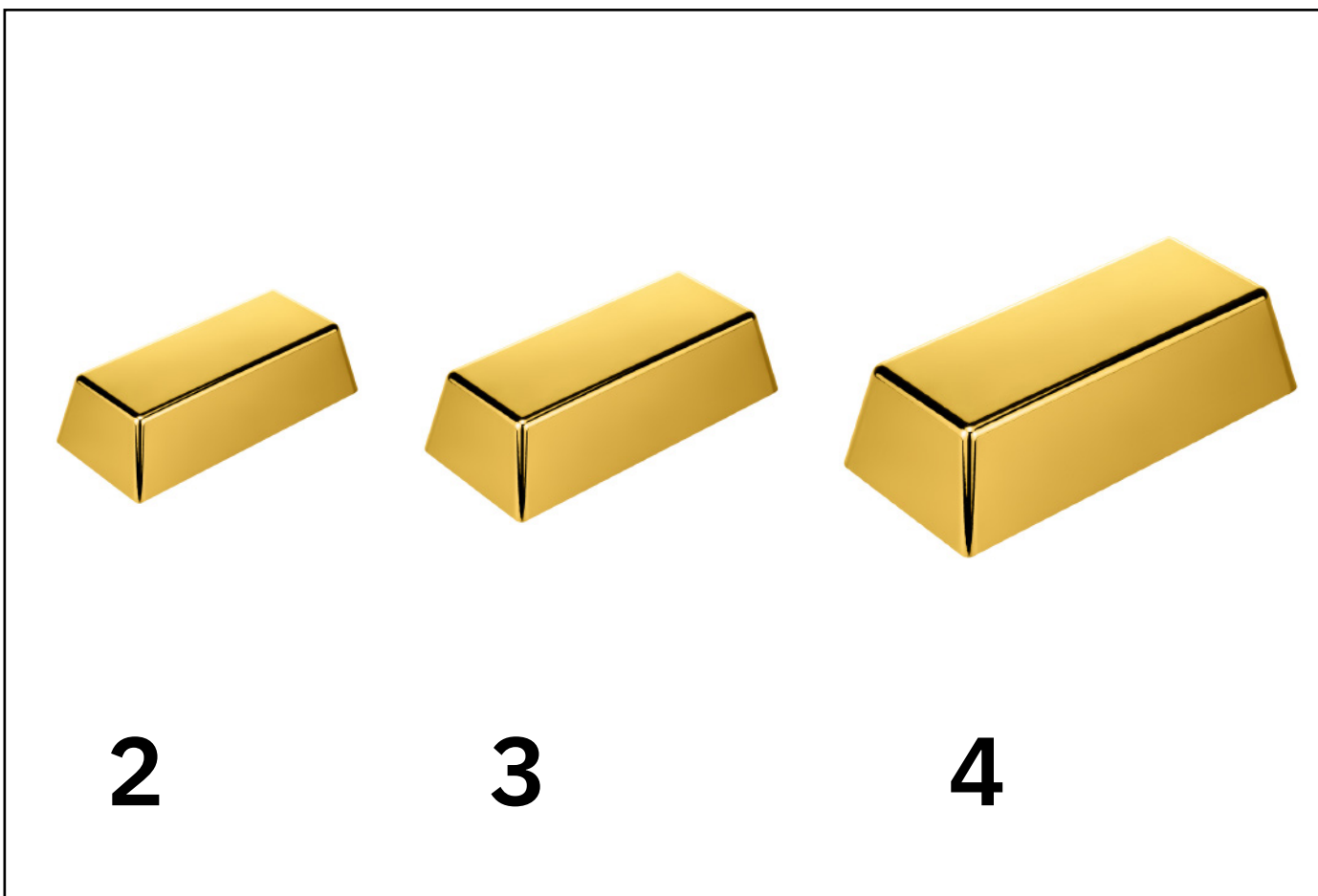
9



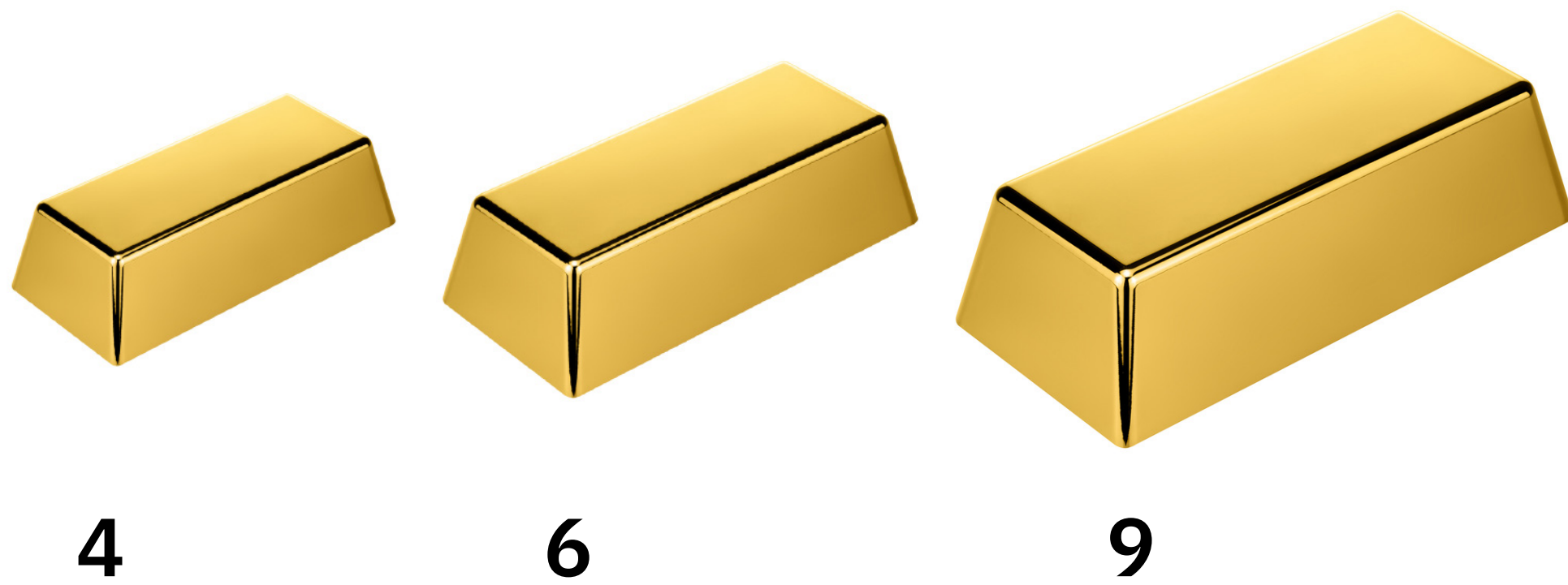
10



# Пример\_



6



10

# Пример\_

6



2



3



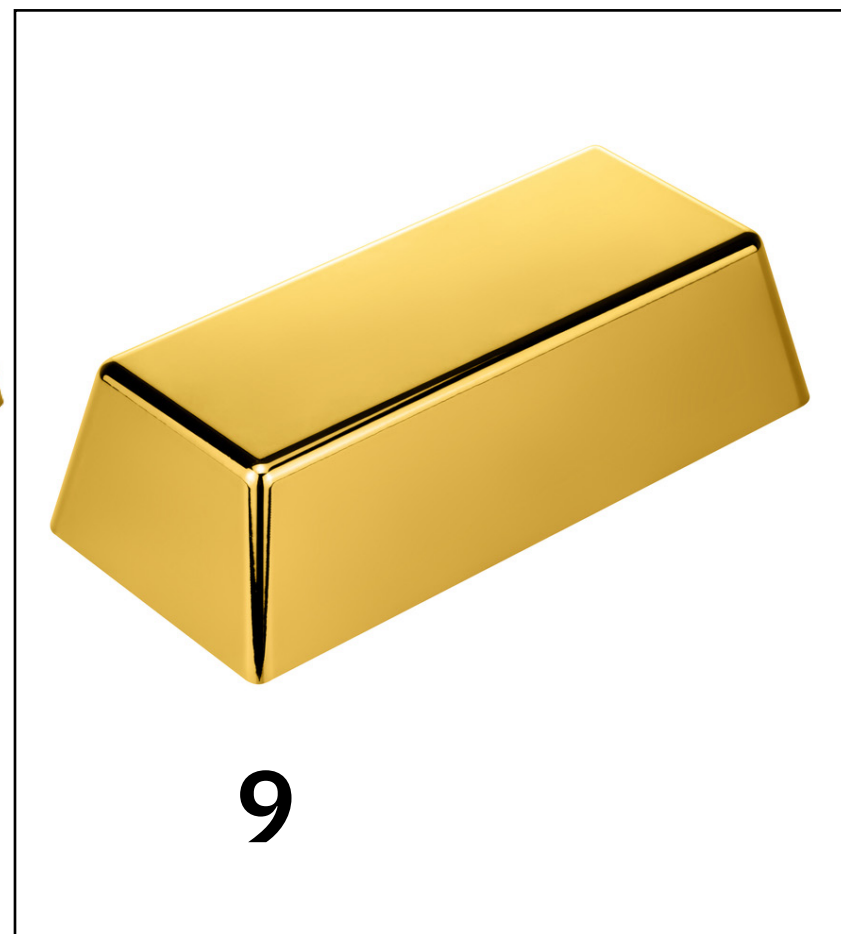
4



4



6



9



10



# Пример\_

6



2



3



4



4



6



9

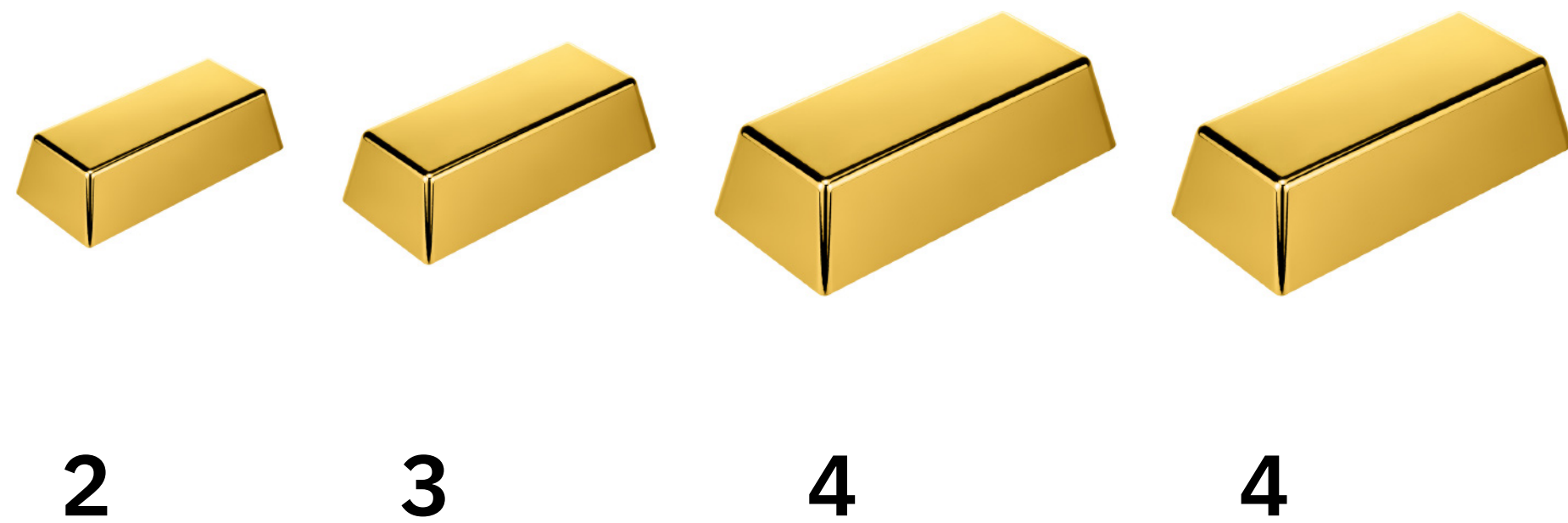


10

# Попытка решить\_

~~$dp_w$  — максимальный вес, который вмещается  
в рюкзак вместимости  $w$~~

$dp_i$  — максимальный вес, который можно набрать,  
используя только первые  $i$  камней

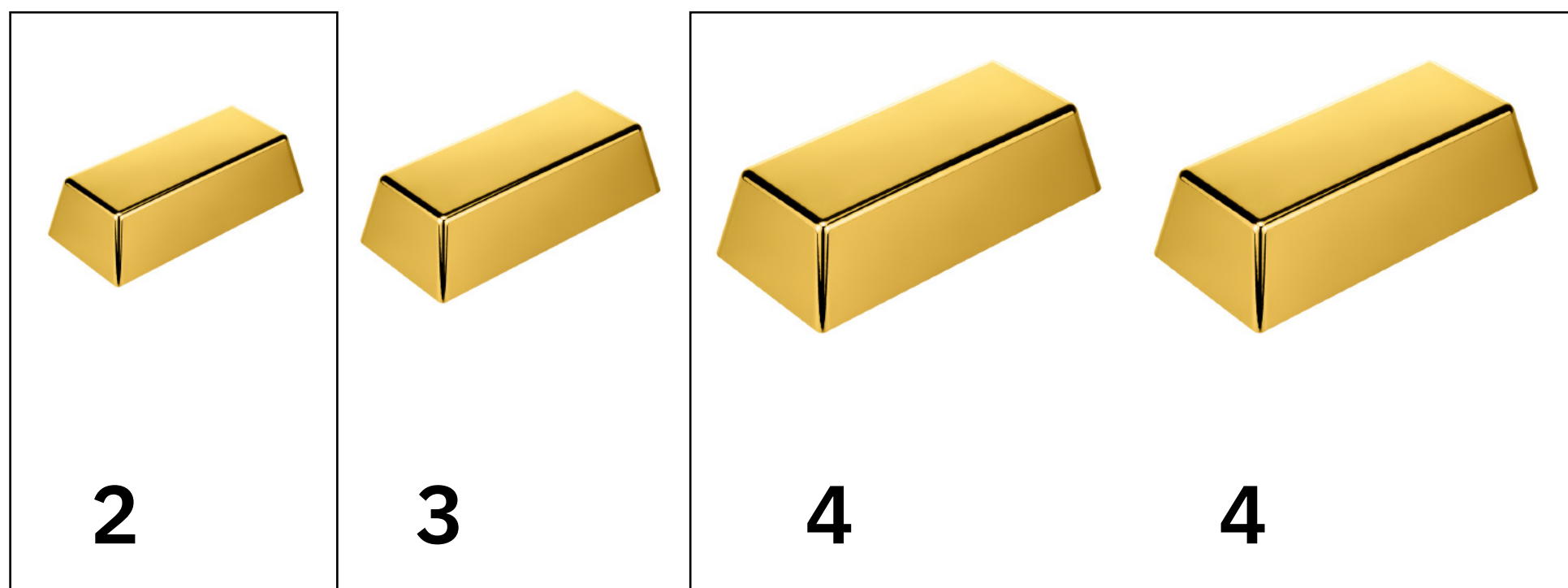




# Попытка решить\_

~~$dp_w$  — максимальный вес, который вмещается  
в рюкзак вместимости  $w$~~

~~$dp_i$  — максимальный вес, который можно набрать,  
используя только первые  $i$  камней~~



# Решение\_

1. Состояние
2. База
3. Формула
4. Порядок
5. Ответ

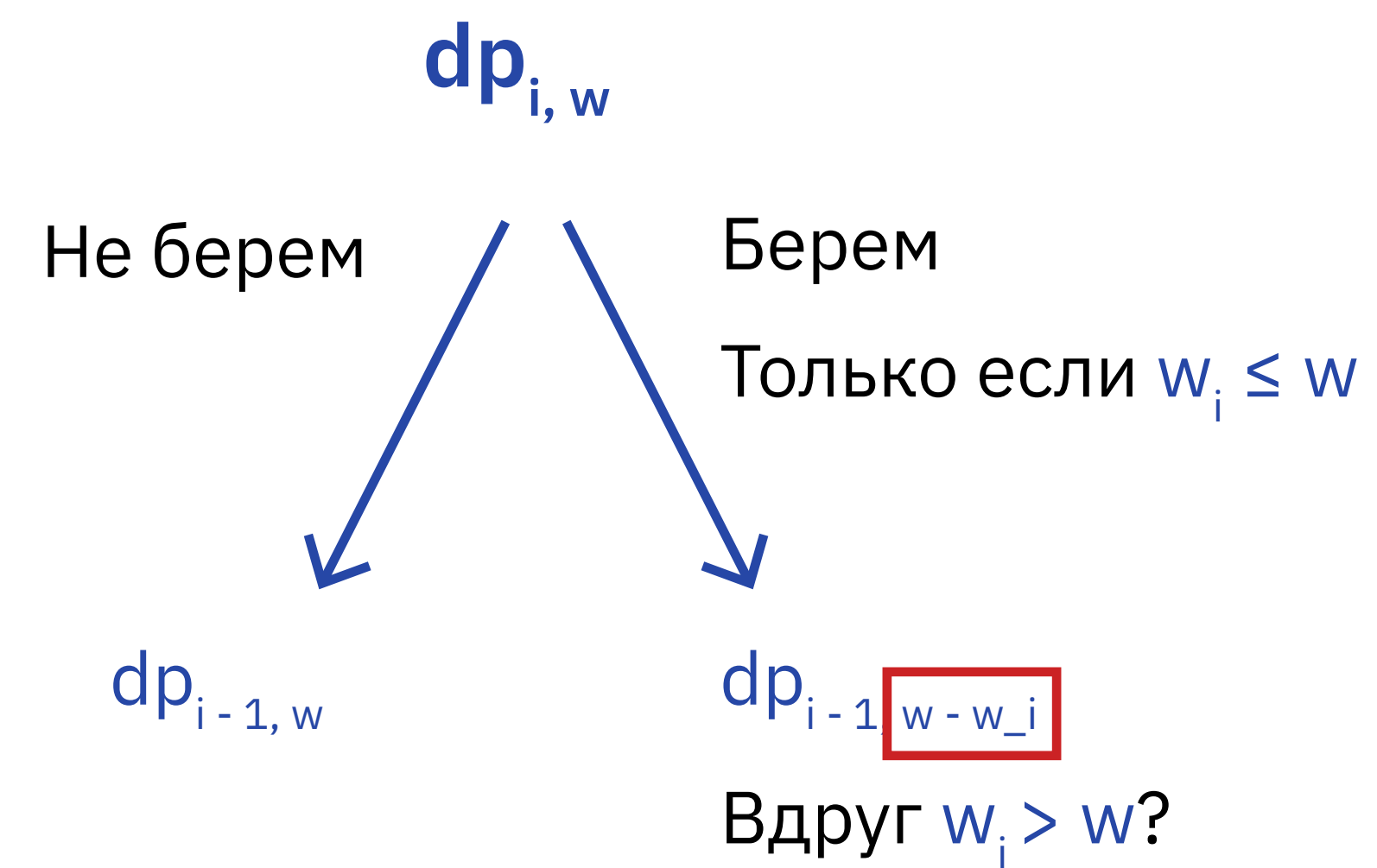
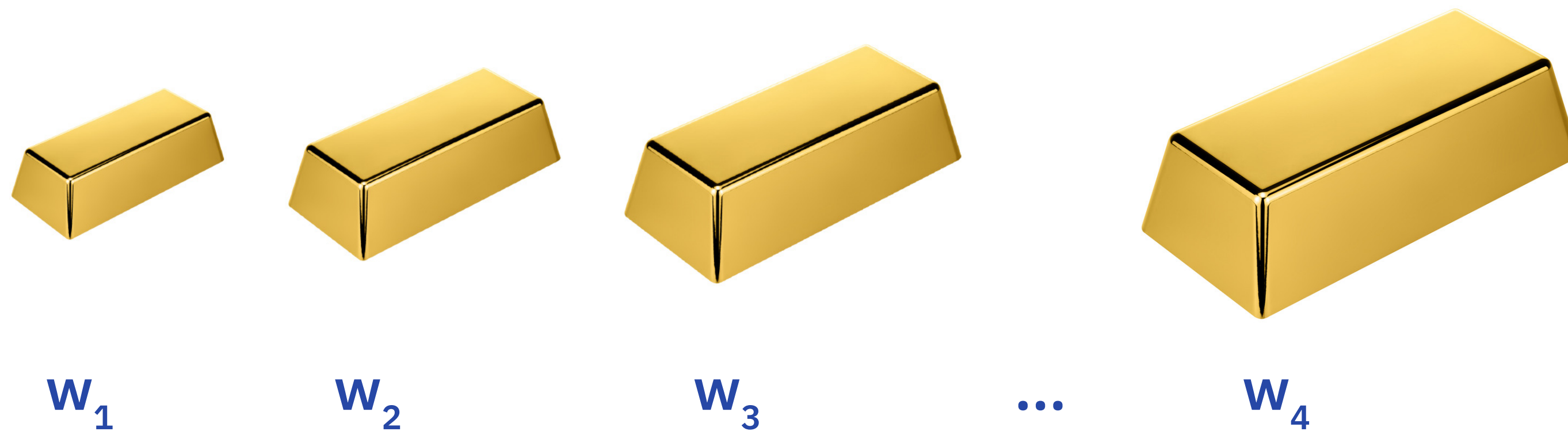


# Решение\_

1. **Состояние:**  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?
2. База
3. Формула
4. Порядок
5. Ответ

# Решение\_

Брать или не брать?





# Решение\_

1. Состояние:  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?

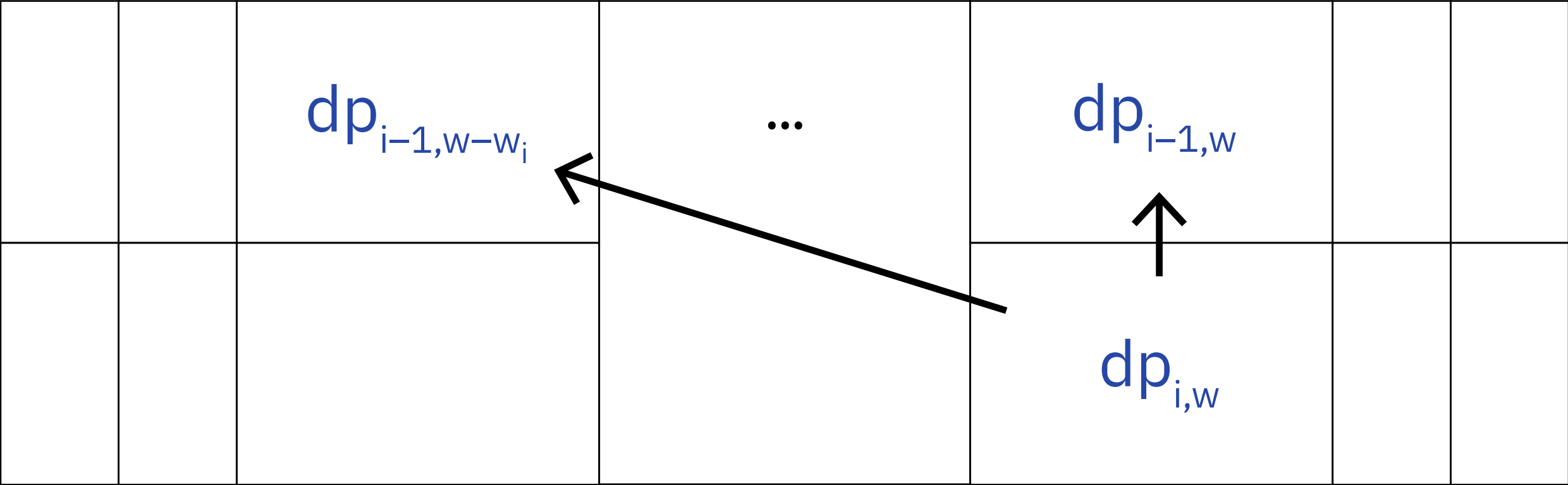
2. База

3. Формула:  $dp_{i,w} = dp_{i-1,w}$  OR  $dp_{i-1,w-w_i}$  если  $w_i \leq w$   
 $dp_{i,w} = dp_{i-1,w}$  иначе

4. Порядок

5. Ответ

# Решение\_





# Решение\_

1. Состояние:  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?

2. База

3. Формула:  $dp_{i,w} = dp_{i-1,w}$  OR  $dp_{i-1,w-w_i}$  если  $w_i \leq w$   
 $dp_{i,w} = dp_{i-1,w}$  иначе

4. Порядок: по возрастанию  $i$ , затем по возрастанию  $w$

5. Ответ

# Решение\_

	0	1	2	3
1				
2				
3				



# Решение\_

	0	1	2	3
0	1	0	0	0
1				
2				
3				

# Решение\_

1. **Состояние:**  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?
2. **База:**  $dp_{0,0} = 1$ ;  $dp_{0,i} = 0$
3. **Формула:**  $dp_{i,w} = dp_{i-1,w}$  OR  $dp_{i-1,w-w_i}$  если  $w_i \leq w$   
 $dp_{i,w} = dp_{i-1,w}$  иначе
4. **Порядок**
5. **Ответ**

# Решение\_

	0	1	2	3
0	1	0	0	0
1	...	...	...	...
2	...	...	...	...
3				?



# Решение\_

	0	1	2	3
0	1	0	0	0
1	...	...	...	...
2	...	...	...	...
3	1	1	0	0

Ответ — 1

# Решение\_

1. **Состояние:**  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?
2. **База:**  $dp_{0,0} = 1$ ;  $dp_{0,i} = 0$
3. **Формула:**  $dp_{i,w} = dp_{i-1,w}$  OR  $dp_{i-1,w-w_i}$  если  $w_i \leq w$   
 $dp_{i,w} = dp_{i-1,w}$  иначе
4. **Порядок:** по возрастанию  $i$ , затем по возрастанию  $w$
5. **Ответ:** наибольшее  $w$ , такое что  $dp_{n,w} = 1$

# Решение\_

```
1  int n, W;
2  cin >> n >> W;
3
4  vector <int> w(n + 1);
5
6  for (int i = 1; i <= n; ++i) {
7      cin >> w[i];
8  }
9
10 vector <vector <int> > dp(n + 1, vector <int> (W + 1, 0));
11 dp[0][0] = 1;
12
13 for (int i = 1; i <= n; ++i) {
14     for (int j = 0; j <= W; ++j) {
15         if (w[i] <= j) {
16             dp[i][j] = dp[i - 1][j] || dp[i - 1][j - w[i]];
17         } else {
18             dp[i][j] = dp[i - 1][j];
19         }
20     }
21 }
22
23 for (int w = W; w >= 0; --w) {
24     if (dp[n][w] == 1) {
25         cout << w << endl;
26         return 0;
27     }
28 }
```



# Итог

1. **Состояние:**  $dp_{i,w}$  — можно ли набрать суммарный вес  $w$ , используя только какие-то из первых  $i$  камней?
2. **База:**  $dp_{0,0} = 1$ ;  $dp_{0,i} = 0$
3. **Формула:**  $dp_{i,w} = dp_{i-1,w}$  OR  $dp_{i-1,w-w_i}$  если  $w_i \leq w$   
 $dp_{i,w} = dp_{i-1,w}$  иначе
4. **Порядок:** по возрастанию  $i$ , затем по возрастанию  $w$
5. **Ответ:** наибольшее  $w$ , такое что  $dp_{n,w} = 1$

# Следующее занятие — более сложная задача о рюкзаке

---