

Задача о рюкзаке

Knapsack problem



Вы грабитель.





Вы грабитель.

Нужно выбрать **вещи**,
которые Вы унесёте...



У всех вещей есть **цены**. Суммарная стоимость украденного должна быть **максимальной**.

Вы грабитель.

Нужно выбрать **вещи**, которые Вы унесёте...



40\$



50\$



60\$



100\$

200\$



У всех вещей есть **цены**. Суммарная стоимость украденного должна быть **максимальной**.

Вы грабитель.

Нужно выбрать **вещи**, которые Вы унесёте...

Но у вещей есть ещё **и вес!**



40\$

3 кг



3 кг

50\$



60\$

5 кг



100\$

12 кг

200\$

18 кг



У рюкзака
ограниченная
вместимость

У всех вещей есть **цены**. Суммарная
стоимость украденного должна быть
максимальной.

Вы грабитель.

Нужно выбрать **вещи**,
которые Вы унесёте...

Но у вещей есть ещё
и **вес!**

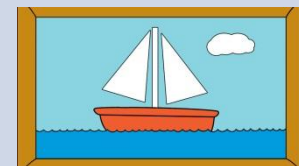


40\$

3 кг

3 кг

50\$



60\$



5 кг



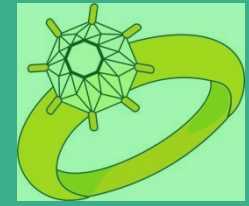
100\$

12 кг

200\$

18 кг

Max: 150\$?



40\$

3 кг

3 кг

50\$



60\$



5 кг



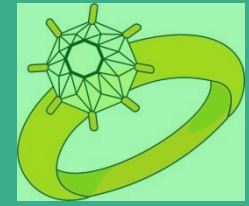
100\$

12 кг

200\$

18 кг

Max: 190\$?

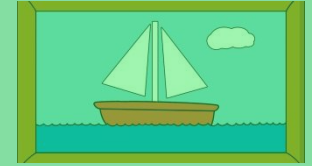


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Max: 200\$?



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Max: 210\$!

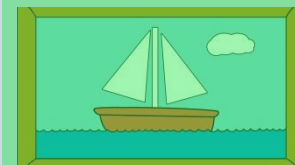


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Max: 210\$!

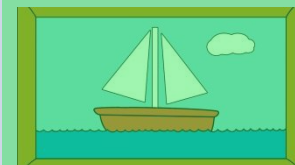


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Формат ввода:

N M

m_1 m_2 ... m_n

c_1 c_2 ... c_n

N – количество вещей

M – вместимость рюкзака

m_i – вес i -ой вещи

c_i – стоимость i -ой вещи



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Формат ввода:

N M

m_1 m_2 ... m_n

c_1 c_2 ... c_n

N – количество вещей

M – вместимость рюкзака

m_i – вес i -ой вещи

c_i – стоимость i -ой вещи

Пример:

5 20

3 3 5 12 18

40 50 60 100 200

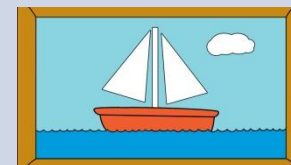


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Формат ввода:

N M

m_1 m_2 ... m_n

c_1 c_2 ... c_n

N – количество вещей

M – вместимость рюкзака

m_i – вес i -ой вещи

c_i – стоимость i -ой вещи

Формат вывода:

Одно число – максимальная стоимость кражи



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Подход 1.

Полный перебор

Сколько всего вариантов?



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$



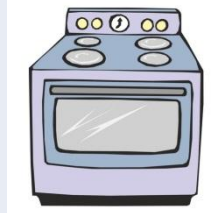

18 кг

Подход 1.





Полный перебор

Сколько всего вариантов?

Каждый предмет мы можем взять или не взять...

+	↙		40\$
-	↘		3 кг
+	↙		50\$
-	↘		3 кг
+	↙		60\$
-	↘		5 кг
+	↙		100\$
-	↘		12 кг
+	↙		200\$
-	↘		18 кг

Подход 1.
Полный перебор
Всего 2^n вариантов

+	↙		40\$
-	↘		3 кг
+	↙		3 кг
-	↘		50\$
+	↙		60\$
-	↘		5 кг
+	↙		100\$
-	↘		12 кг
+	↙		200\$
-	↘		18 кг





Подход 1.

Полный перебор

- *Даёт верный результат*
- *Оооочень долго*

(при $n = 100$ суперкомпьютер будет вычислять несколько тысяч лет)



+	↙		40\$
-	↘		3 кг
+	↙		50\$
-	↘		3 кг
+	↙		60\$
-	↘		5 кг
+	↙		100\$
-	↘		12 кг
+	↙		200\$
-	↘		18 кг

Подход 2.

Брать самую дорогую вещь

- *Не всегда будет давать максимальный ответ
(уже в нашем примере не работало)*
- *Приведите пример из трёх вещей*



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Подход 3.

Рассчитать плотность каждой

вещи: c_i/m_i .

Брать вещи, в порядке убывания
плотности, пока влезают.



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Подход 3.

Рассчитать плотность каждой

вещи: c_i/m_i .

Брать вещи, в порядке убывания плотности, пока влезают.

- Тоже не всегда будет давать правильный ответ.
(приведите пример)



	40\$
3 кг	50\$
	
3 кг	50\$
	60\$
5 кг	
	100\$
12 кг	
	200\$
	18 кг

Подходы 2-3.

Последние два алгоритма назывались *жадными*.

Для данной задачи они не подходят.



	40\$
3 кг	50\$
	
3 кг	50\$
	60\$
5 кг	
	100\$
12 кг	
	200\$
	18 кг

Подход 4.

Метод динамического программирования.



	40\$
3 кг	50\$
	
	60\$
	5 кг
	100\$
	12 кг
	200\$
	18 кг

Подход 4.

Метод динамического программирования.

Заведём табличку $(N + 1) \times (M + 1)$



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

18 кг

Подход 4.

Метод динамического программирования.

Заведём табличку $P: (N + 1) \times (M + 1)$

Будем считать, что все вещи пронумерованы.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2																						
3																						
4																						
5																						

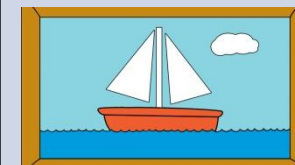


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

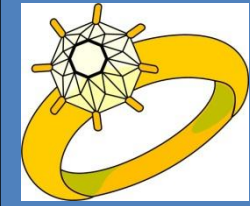
12 кг

200\$

18 кг

В $P[i][j]$ будем хранить максимальную сумму, которую можно заработать, выбирая среди первых i вещей и имея рюкзак вместимостью j кг.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2																						
3																						
4																						
5																						



40\$

3 кг



3 кг

50\$



60\$

5 кг



100\$

12 кг

200\$

18 кг

В $P[i][j]$ будем хранить максимальную сумму, которую можно заработать, выбирая среди первых i вещей и имея рюкзак вместимостью j кг.

Что будет здесь?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2																						
3																						
4																						
5																						



40\$

3 кг



3 кг

50\$



60\$

5 кг



100\$

12 кг

200\$

18 кг

В $P[i][j]$ будем хранить максимальную сумму, которую можно заработать, выбирая среди первых i вещей и имея рюкзак вместимостью j кг.

Выбираем среди первых двух вещей

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2					↓																	
3																						
4																						
5																						

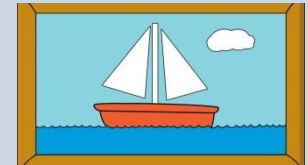


40\$

3 кг

3 кг

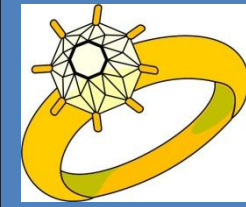
50\$



В $P[i][j]$ будем хранить максимальную сумму, которую можно заработать, выбирая среди первых i вещей и имея рюкзак вместимостью j кг.

Выбираем среди первых двух вещей.
Рюкзак вместимостью 4 кг.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2																						
3																						
4																						
5																						



40\$

3 кг

3 кг

50\$



В $P[i][j]$ будем хранить максимальную сумму, которую можно заработать, выбирая среди первых i вещей и имея рюкзак вместимостью j кг.



40\$

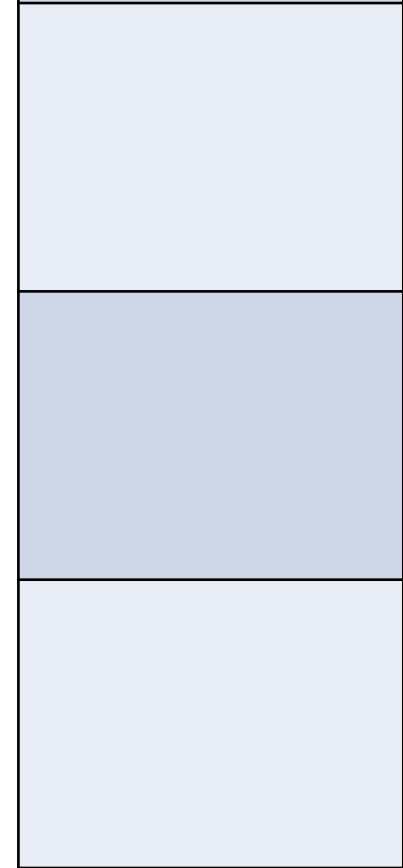
3 кг

3 кг

50\$



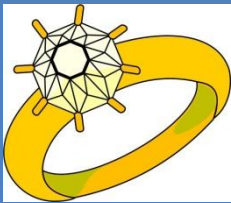


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2					50																	
3																						
4																						
5																						



1. Сначала заполним очевидное:
нулевой столбец и нулевую строку.

Как?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0																						
1																						
2																						
3																						
4																						
5																						

	40\$	3 кг
	50\$	3 кг
	60\$	5 кг
	100\$	12 кг
	200\$	

1. Сначала заполним очевидное:
нулевой столбец и нулевую строку.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
3	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$

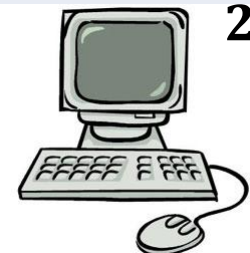


5 кг



100\$

12 кг



200\$

1. Сначала заполним очевидное: нулевой столбец и нулевую строку.
2. Будем заполнять оставшуюся часть сверху-вниз, слева-направо.


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0																						
2	0																						
3	0																						
4	0																						
5	0																						



40\$
3 кг




3 кг **50\$**



60\$
5 кг



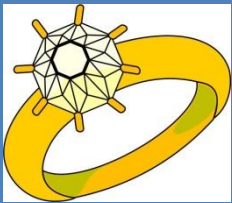
100\$
12 кг



200\$

1. Сначала заполним очевидное: нулевой столбец и нулевую строку.
2. Будем заполнять оставшуюся часть сверху-вниз, слева-направо, выражая каждый следующий ответ через предыдущие.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
3	0																					
4	0																					
5	0																					



40\$
3 кг

3 кг 50\$



60\$




5 кг

100\$



12 кг

200\$



Допустим, мы вычислили всё до i -ой строки и j -го столбца.

Сейчас вычисляем $P[i][j]$.

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					

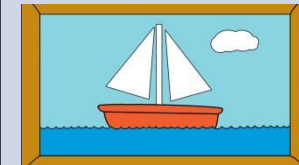


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



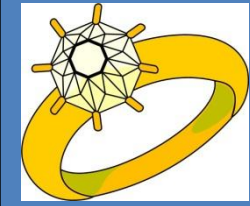
200\$

Есть два варианта



Взять вещь №*i*
(последнюю)

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																				
2	0																				
i	0																				
4	0																				
5	0																				



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



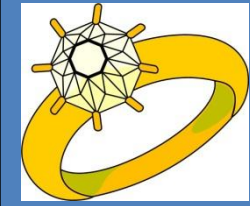
200\$

Есть два варианта

Взять вещь №**i**
(последнюю)

Не брать вещь
№**i**

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$

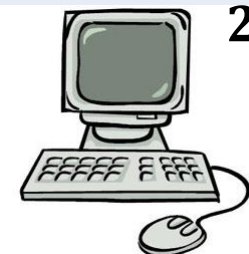


5 кг



100\$

12 кг



200\$

Есть два варианта

Взять вещь №*i*
(последнюю)

Не брать вещь
№*i*

Вычислим эти значения и выберем **наибольшее**

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

Есть два варианта

Взять вещь №i

Не брать вещь
№i

Только, если влезает.
Что это значит?

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																				
2	0																				
i	0																				
4	0																				
5	0																				



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Только, если влезает

$$w_i \leq j$$

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																				
2	0																				
i	0																				
4	0																				
5	0																				



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$


Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*


Мы точно
зарабатываем: c_i

	0	1	2	3	4	5	6	<i>j</i>	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
<i>i</i>	0																					
4	0																					
5	0																					



40\$
3 кг

3 кг 50\$



60\$



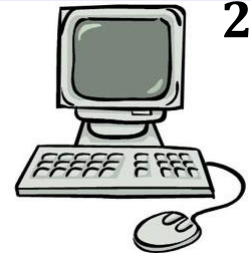
5 кг

100\$



12 кг

200\$



Есть два варианта

Взять вещь №i

Не брать вещь №i

Плюс, у нас осталось еще свободное место...

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

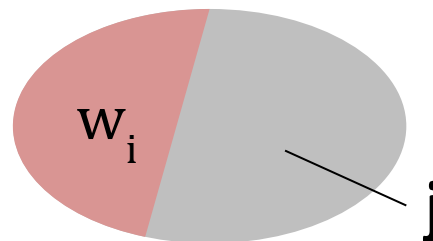
Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Плюс, у нас осталось еще свободное место...

Сколько?



	0	1	2	3	4	5	6	<i>j</i>	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
<i>i</i>	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



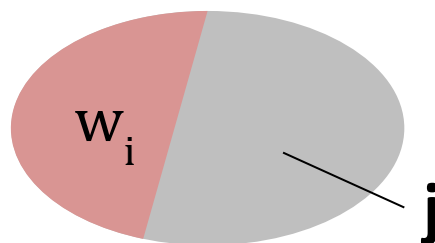
200\$

Есть два варианта

Взять вещь №**i**

Не брать вещь №**i**

$$j - w_i$$



	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					

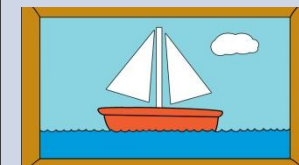


40\$

3 кг

3 кг

50\$



60\$

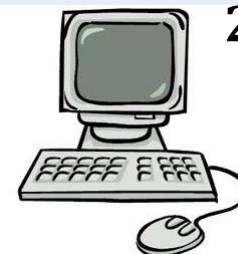


5 кг



100\$

12 кг



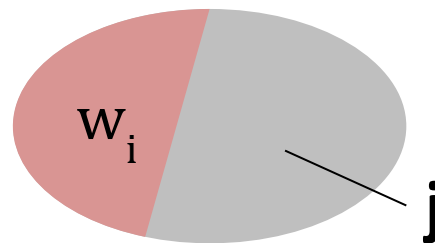
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

*Как его оптимально
заполнить?*



	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0																						
2	0																						
i	0																						
4	0																						
5	0																						



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



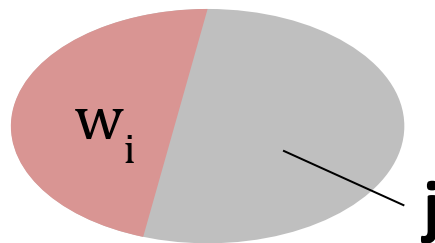
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Таблица *P* знает
ответ на этот
вопрос!

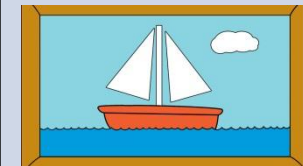


	0	1	2	3	4	5	6	<i>j</i>	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
<i>i</i>	0																					
4	0																					
5	0																					



40\$

3 кг



3 кг

50\$



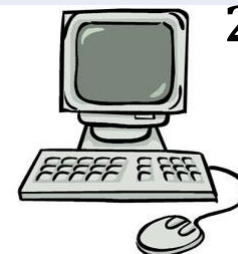
60\$

5 кг



100\$

12 кг



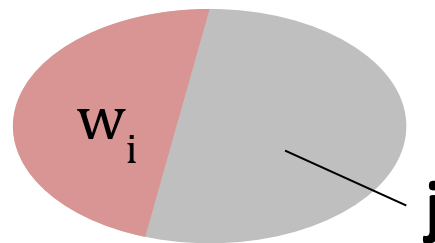
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

В ячейке P[...][...] уже лежит лучшая сумма

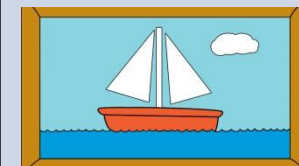


	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0																						
2	0																						
i	0																						
4	0																						
5	0																						



40\$

3 кг



3 кг

50\$



60\$

5 кг



100\$

12 кг



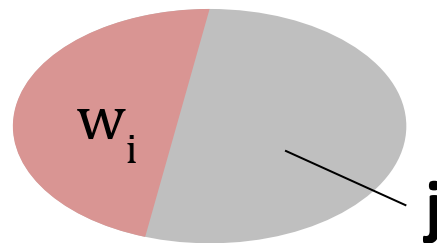
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

В ячейке $P[...][j-w_i]$ уже лежит лучшая сумма



	0	1	2	3	4	5	6	<i>j</i>	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
<i>i</i>	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



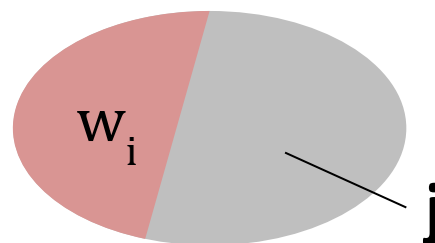
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

В ячейке $P[i-1][j-w_i]$ уже лежит лучшая сумма



	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



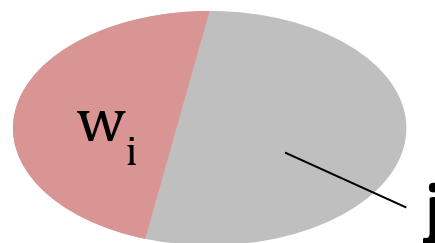
200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Итого:
 $c_i + P[i-1][j-w_i]$,
 если влезет



	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

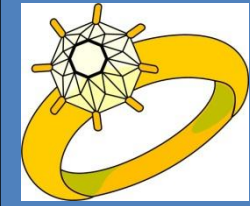
Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Итого:
 $c_i + P[i-1][j-w_i]$,
 если влезет

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг

200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Итого:
 $c_i + P[i-1][j-w_i]$,
 если влезет

В ячейке $P[...][...]$
 уже есть ответ на
 эту задачу

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0																					
i	0																					
4	0																					
5	0																					



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Итого:
 $c_i + P[i-1][j-w_i]$,
 если влезет

В ячейке $P[i-1][...]$
 уже есть ответ на
 эту задачу

	0	1	2	3	4	5	6	j	8	9	10	11	12	13	14	15	16	17	18	19	20		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0																						
2	0																						
i	0																						
4	0																						
5	0																						



40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$

Есть два варианта

Взять вещь №*i*

Не брать вещь №*i*

Итого:
 $c_i + P[i-1][j-w_i]$,
 если влезет

В ячейке $P[i-1][j]$
 уже есть ответ на
 эту задачу

	0	1	2	3	4	5	6	<i>j</i>	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0																					
2	0			x				x														
<i>i</i>	0																					
4	0																					
5	0																					

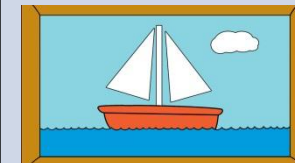


40\$

3 кг

3 кг

50\$



60\$



5 кг



100\$

12 кг



200\$