

# Задача об ограниченном ранце.

Абрамов З.И.

Швецов Г.А.

группа ФН2-42Б

18 июня 2022 г.



## Постановка задачи

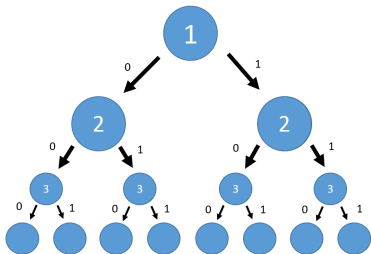
Пусть имеется  $n$  типов предметов. Каждый тип предмета  $i$  характеризуется весом  $w_i$  и стоимостью  $c_i$  одного предмета и количеством предметов  $k_i$  данного типа. Также имеется рюкзак вместимости  $W$ .

Требуется собрать набор с максимальной полезностью таким образом, чтобы он имел вместимость не больше  $W$ . При этом количество предметов типа  $i$  не должно превышать  $k_i$ .

В математической форме:

$$\begin{aligned} \sum_{i=1}^n c_i x_i &\rightarrow \max \\ \sum_{i=1}^n w_i x_i &\leq W \\ \forall i \in \{1, \dots, n\} \quad x_i &\in \{0, \dots, k_i\} \end{aligned}$$

# Метод ветвей и границ



Дерево полного перебора,  
соответствующее поиску решения  
для трех предметов.

Метод ветвей и границ  
является вариацией метода  
полного перебора с той  
разницей, что исключаются  
заведомо неоптимальные ветви  
дерева полного перебора.

## Метод ветвей и границ



⇒ новое решение

## Динамическое программирование


Подход динамического программирования состоит в том, что если при решении исходной задачи часто решаются одинаковые подзадачи, то имеет смысл сохранять решение таких подзадач, сократив тем самым количество вычислений.



# Динамическое программирование

В качестве таких подзадач будем решать задачу о ранце для первых  $n' < n$  предметов и рюкзака вместимости  $W' < W$ .

$\begin{smallmatrix} W' \\ n' \end{smallmatrix}$	1	2	3	4	5	6
1						
2						
3						
4						



$$W = 6, \quad n = 4;$$

$$n' = 3, \quad W' = 5;$$

$$w_3 = 2, \quad k_3 = 2;$$

$$dp[3, 4] = \max_{x_i=0,1,2} (x_i \cdot c_3 + dp[2, 4 - x_i \cdot 2]).$$

## Пример работы

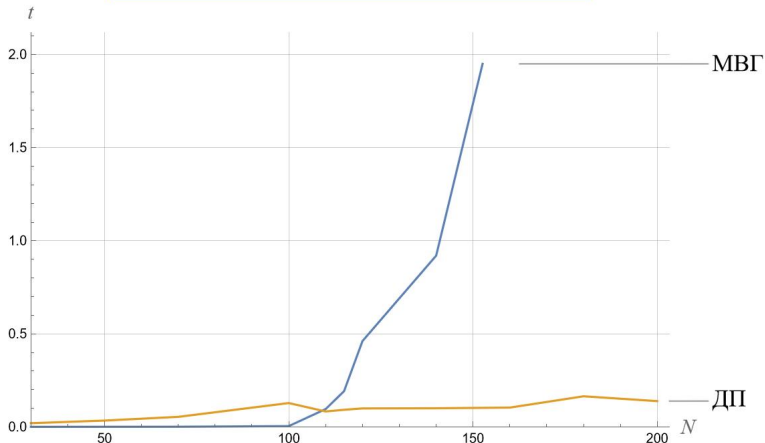
Пример с двумя решениями при  $\text{max\_weight} = 1000$ .

Исходные данные			
item	count	weight	cost
1	6	54	8
2	5	63	12
3	4	55	2
4	3	63	7
5	8	97	6
6	6	26	11
7	4	91	16
8	7	42	2
9	5	8	18
10	6	18	3

Решение 1			
item	count	max_cost	sum_weight
10	6	298	983
7	4		
2	5		
6	6		
9	5		
Решение 2			
item	count	max_cost	sum_weight
1	2	298	1000
10	6		
7	3		
2	5		
6	6		
9	5		

## Время исполнения методов на C++

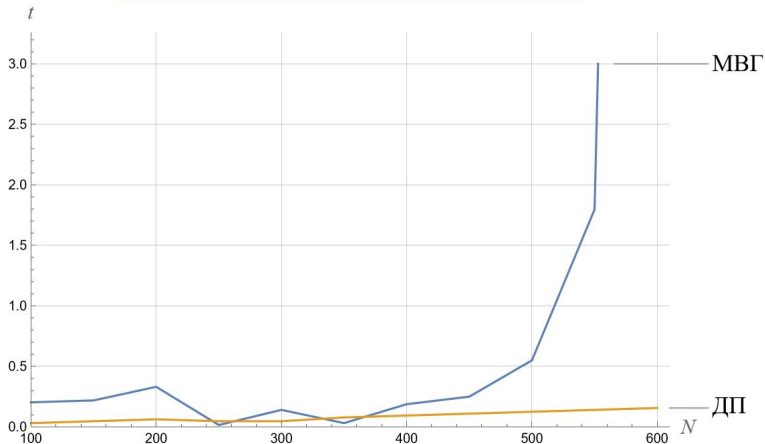
`max_weight=10000, count<=10, weight<=50, cost<=100`





# Время исполнения методов на Wolfram Mathematica

`max_weight=500, count<=10, weight<=50, cost<=100`



Спасибо за внимание!