МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тверской государственный технический университет»

(ТвГТУ)

Кафедра “Программного обеспечения”

**Курсовая работа**

по дисциплине «Теория вычислительных процессов»

Тема: «Задача о рюкзаке»

Выполнил: студент группы

ПИН 17.06

Иванов Р. В.

Проверил:

Калабин А.Л

Тверь 2021

Оглавление

[Цель работы 3](#_Toc68436649)

[Постановка задач 3](#_Toc68436650)

[Описание точного алгоритма 3](#_Toc68436651)

[Описание приближенного алгоритма 4](#_Toc68436652)

[Зависимость времени решения от размерности входных данных для точного решения 4](#_Toc68436653)

[Зависимость времени решения от размерности входных данных для приближенного решения 5](#_Toc68436654)

[Зависимость точности решения от размерности входных данных для приближенного решения 5](#_Toc68436655)

[Выводы по проделанной работе 6](#_Toc68436656)

[Литература 6](#_Toc68436657)

# Цель работы

1. Решение задачи о рюкзаке методом полного перебор
2. Решение задачи о рюкзаке приближенным алгоритмом: случайный перебор
3. Построить зависимости времени решения от размерности входных данных.
4. Сравнить эту зависимость с зависимостью для точного решения.
5. Построить зависимость точности решения от размерности входных данных

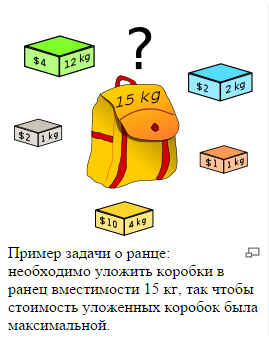
# Введение

Задача о ранце (рюкзаке) (англ. Knapsack problem) — одна из NP-полных задач комбинаторной оптимизации. Название своё получила от максимизационной задачи укладки как можно большего числа нужных вещей в рюкзак при условии, что общий объём (или вес) всех предметов, способных поместиться в рюкзак, ограничен. Задачи о загрузке (о рюкзаке) и её модификации часто возникают в экономике, прикладной математике, криптографии, генетике и логистике для нахождения оптимальной загрузки транспорта (самолёта, поезда, трюма корабля) или склада.

В общем виде задачу можно сформулировать так: из заданного множества предметов со свойствами «стоимость» и «вес», требуется отобрать некое число предметов таким образом, чтобы получить максимальную суммарную стоимость при одновременном соблюдении ограничения на суммарный вес

# Постановка задач

Пусть имеется набор предметов, каждый из которых имеет два параметра — масса и ценность. Также имеется рюкзак определённой грузоподъёмности. Задача заключается в том, чтобы собрать рюкзак с максимальной ценностью предметов внутри, соблюдая при этом ограничение рюкзака на суммарную массу.



# Полный перебор

Название метода говорит само за себя. Чтобы получить решение нужно перебрать все возможные варианты загрузки. Здесь мы будем рассматривать такую постановку задачи. В рюкзак загружаются предметы N различных типов (количество предметов каждого типа не ограничено), каждый предмет типа I имеет вес Wi и стоимость Pi , i=(1,2..N). Требуется определить максимальную стоимость груза вес, которого не превышает W. Очевидна простая рекурсивная реализация данного подхода Рис 1.

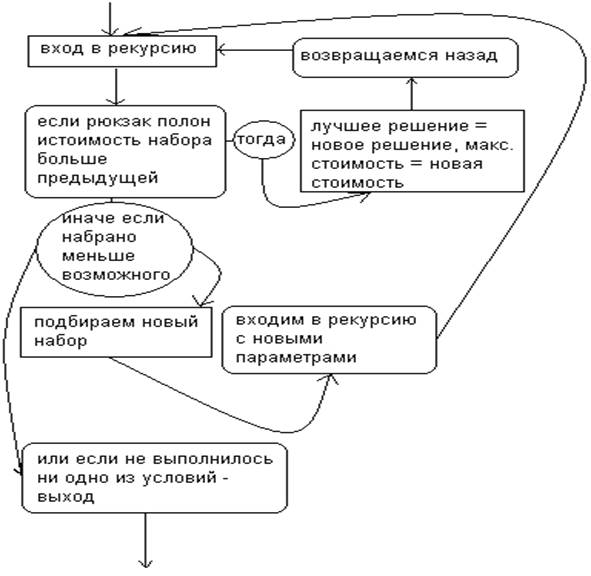


Рис 1.

Временная сложность данного алгоритма равна O(N!). Алгоритм имеет сложность факториал и может работать лишь с небольшими значениями N. С ростом N, число вариантов очень быстро растет, и задача становится практически неразрешимой методом полного перебора. На рис 2. показано дерево перебора, дерево имеет 4 уровня. В каждом кружочке показан вес предмета, корень дерева – нулевой вес, то есть, когда рюкзак пуст. Первый предмет можно выбрать четырьмя способами, второй – тремя, третий – двумя, а дальше можем взять только один оставшийся предмет.

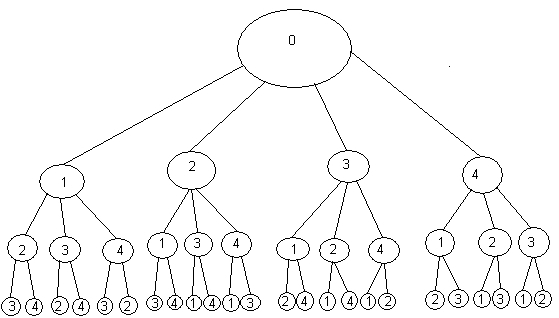


Рис 2.

# Описание случайного перебора

Чтобы получить решение нужно заполнить полностью заполнить рюкзак, случайными предметами.

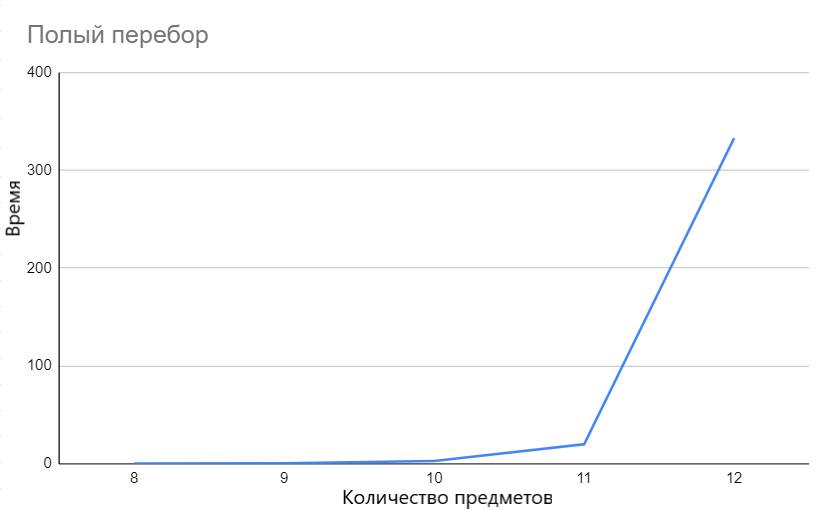
На каждой итерации мы берем случайный предмет из списка доступный предметов, если он еще не находится в рюкзаке и при его добавлении масса всех предметов в рюкзаке не превысит максимально допустимую, то мы добавляем этот предмет в рюкзак, вычисление останавливаются после того, как не остается предметов, которые мы могли бы добавить в рюкзак.

Временная сложность данного алгоритма линейная O(N) и зависит от размера рюкзака.

# Зависимость времени решения от размерности входных данных для полного перебора

Тестирование производилось на рюкзаке размером 200 и количестве предметов 5, 10, 11, 12.

Результат:

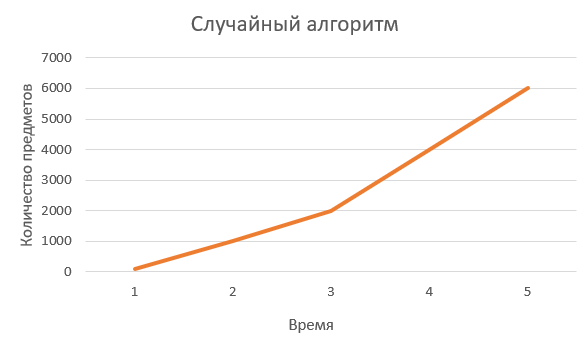


Время в секундах

Время очень быстро растет с количеством входных данных.

С количеством предметов> 11 вычисление занимает ~4 минуты.

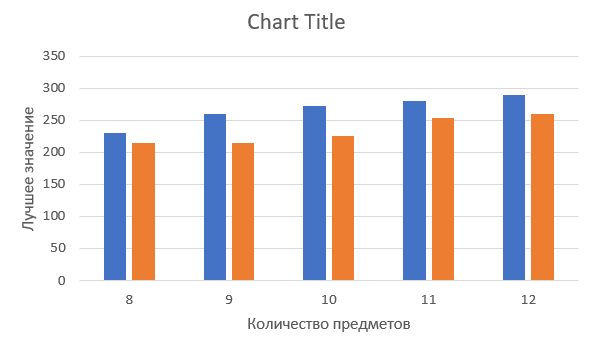
# Зависимость времени решения от размерности входных данных для случайного перебора



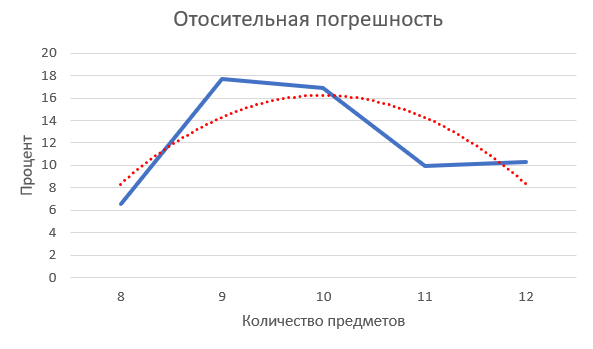
Время в миллисекундах

В случайном переборе время зависит от количества входных параметров и размера рюкзака.

# Зависимость точности решения от размерности входных данных для приближенного решения







Как можно увидеть, результаты случайного алгоритма сильно отличаются от эталонных результатов полного перебора.

# Выводы по проделанной работе

Полны перебор отличается быстрым ростом времени выполнения при изменении входных данных, но дает правильное решение в любых случаях.

Случайный перебор практически всегда выдает не оптимальное решение, но считает очень быстро на любых входных данных.

# Литература

1 Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. — СПб.: Наука и Техника, 2006. — 320