# Постановка задачи

## Задача о рюкзаке

Задача о рюкзаке является одной из наиболее известных задач комбинаторной оптимизации. В данной работе рассматривается «0-1 задача» со следующей постановкой.

Имеется множество предметов, каждый из которых имеет два параметра – вес и стоимость. Необходимо найти такое подмножество с суммарным весом, не превышающим заданного (веса рюкзака), что суммарная стоимость является наибольшей среди всех возможных.

В работе на множество входных данных накладываются ограничения:

1. максимальный вес рюкзака
2. количество предметов

Задача относится к классу NP-трудных [1].

## Разрабатываемое приложение

В рамках работы требуется разработать консольное приложение для решения задачи о рюкзаке, соответствующее следующим требованиям.

Функциональные требования:

1. реализация алгоритмов, описанных в разделе 4;
2. измерение времени работы каждого алгоритма в соответствии с разделом 2;
3. вывод результатов применения алгоритмов к входным данным, включающий среднее время расчета и параметры рюкзака (вес, стоимость, номера предметов).

Требования к входным и выходным данным:

1. входной файл с именем input\_knap.txt находится в папке с исполняемым файлом перед запуском программы;
2. выходной файл с именем output\_knap.txt создается после выполнения программы в папке с исполняемым файлом;
3. при наличии файла с именем output\_knap.txt в папке с исполняемым файлом происходит перезапись содержимого;
4. структура входного файла по строкам (строки 2-5 повторяются N раз):
5. количество тестов N;
6. максимальный вес рюкзака;
7. количество предметов;
8. веса предметов (через пробел);
9. стоимости предметов (через пробел);
10. структуры выходного файла по строкам (строки 1-6 повторяются N раз):
11. номер теста;
12. наименование метода, среднее время расчета в тактах процессора;
13. вес и стоимость рюкзака;
14. номера предметов в рюкзаке;
15. вес предметов в рюкзаке;
16. стоимость предметов в рюкзаке;
17. вывод дублируется в консоль.

Технические требования:

1. приложение разрабатывается совместимо с архитектурой x86;
2. асимптотическая эффективность алгоритмов по времени и требуемой памяти соответствует разделу 4;
3. память освобождается после выполнения серии тестов.

# Инструменты измерения времени

Для измерения времени выполнения алгоритмов используется счетчик Time Stamp Counter (TSC), описанный в справочном руководстве для разработчиков программного обеспечения по архитектуре Intel [2].

Компилятор Microsoft Visual C++ предоставляет встроенную функцию в заголовочном файле intrin.h, MinGW – в заголовочном файле x86intrin.h.

Извлечение текущего значения счетчика TSC происходит непосредственно перед выполнением первой инструкции алгоритма и после последней инструкции алгоритма. Разностью этих значений является временем выполнения алгоритма в циклах процессора. Перевод в единицы измерения, базирующиеся на секундах, не производится в связи с переменной частотой современных процессоров и особенностями планировщиков ОС. Для примерной справки возможно примерно приравнять время цикла к наносекунде.

# Описание плана эксперимента

Описание

# Описание алгоритмов

Описание

# Методы решения

Методы

# Сравнительный анализ методов

Сравнение

# Использованные источники

1. Silvano Martello, Paolo Toth (1990). Knapsack Problems: Algorithms and computer interpretations.
2. Intel® 64 and IA-32 Architectures Software Developer’s Manual (2015).