## 1. Умножение двух чисел.

1. Найти с помощью рекурсивного алгоритма произведение двух чисел.
2. Найти с помощью умножения Карацубы произведение двух чисел.

**Тестовые примеры:** для этой задачи вы можете создавать тестовые примеры, просто вставляя числа в калькулятор. Например, 99 999 * 9 999 равно 999 890 001.
**Задача:** каково произведение
3141592653589793238462643383279502884197169399375105820974944592 и
2718281828459045235360287471352662497775724270957799666?

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 2. Умножение матриц

1. Найти с помощью рекурсивного алгоритма произведение двух матриц.
2. Найти с помощью алгоритма Штрассена произведение двух матриц.

В этой задаче формат файла:
[n-размерность]
[1-ая строка 1-ой матрицы]
[2-ая строка 1-ой матрицы]
…
[n-ая строка 1-ой матрицы]
[1-ая строка 2-ой матрицы]
[2-ая строка 2-ой матрицы]
…
[n-ая строка 2-ой матрицы]
**Тестовый пример:** этот файл
**Набор данных испытания**: этот файл

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 3. Подсчет инверсий

1. Найти количество инверсий в массиве методом «грубой силы».
2. Найти количество инверсий в массиве, используя стратегию «разделяй и властвуй»

**Проверка работоспособности:** во-первых, убедитесь, что ваши алгоритмы подсчитывают 0 инверсий для отсортированного массива и n(n-1)/2 инверсий для обратно отсортированного массива (например, 28 инверсий для [ 8 7 6 5 4 3 2 1 ]).
**Тестовый пример:** этот файл
содержит 10 целых чисел, представляющих массив из 10 элементов.

**Набор данных испытания**: этот файл содержит все целые числа от 1 до 100 000 (включительно) в определенном порядке, без повторения целых чисел. i-я строка файла указывает i-ю запись массива. Сколько инверсий имеет этот массив?

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 4. Нахождение ближайшей пары точек

1. Найти ближайшую пару точек методом «грубой силы».
2. Найти ближайшую пару точек, используя стратегию «разделяй и властвуй»

**Тестовый пример**: 8 точек с координатами (1, 1), (2, 5), (1 ,9), (4, 3), (6, 4), (5,7), (8, 2), (8, 9)

**Набор данных испытания**: равномерно выбрать 1000 точек из квадрате $[0;\ 100] \times [0;\ 100]$

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 5. Быстрая сортировка

Реализовать алгоритм QuickSort.

Для нескольких входных массивов определите количество сравнений, выполняемое следующими ниже реализациями подпрограммы ChoosePivot.

1. В качестве опорного всегда используйте первый элемент.
2. В качестве опорного всегда используйте последний элемент.
3. В качестве опорного используйте случайный элемент. В этом случае выполните алгоритм 10 раз с заданным входным массивом и усредните результаты.
4. В качестве опорного элемента используйте *медиану из трех*, т.е. предварительно выбирать медиану из первого, среднего и последнего элементов массива.

**Тестовый пример № 1:** этот файл содержит 10 целых чисел, представляющих массив из 10 элементов.

**Тестовый пример № 2**: этот файл содержит 100 целых чисел, представляющих массив из 100 элементов.

**Набор данных испытания:** этот файл содержит все целые числа от 1 до 10 000 (включительно) в определенном порядке, без повторения целых чисел. i-я строка файла указывает i-ю запись массива. Сколько сравнений QuickSort делает с этими входными данными, если первый элемент всегда выбирается в качестве опорного? Если последний элемент всегда выбирается в качестве опорного? Если в качестве точки опоры всегда выбирается *медиана из трех*?

Привести и обосновать асимптотическую оценку временной сложности алгоритма в случае рандомизированного выбора опорного элемента.

## 6. Нахождение медианы

1. Найти медиану массива с помощью сортировки.
2. Найти медиану массива с помощью процедуры рандомизированного выбора RSelect

Использовать два тестовых примера, затем сформировать массив из первых цифр в записи числа пи (см. ниже). Сравнить временные затраты для задачи 1 и для задачи 2.

**Тестовый пример № 1:** этот файл содержит 10 целых чисел, представляющих массив из 10 элементов. Найти медиану (то есть 5-й наименьший элемент)?

**Тестовый пример № 2:** этот файл содержит 100 целых чисел, представляющих массив из 100 элементов. . Найти медиану (т. е. статистику 50-го порядка)?

**Набор данных задачи:** сформируйте массив из 1000 элементов, в котором первый элемент — это первые 10 цифр в десятичной записи числа пи, второй элемент — следующие 10 цифр числа пи и так далее. (Цифры числа пи доступны здесь .) Что такое медиана массива? Есть ли в этом массиве повторяющиеся элементы?

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 7. Нахождение медианы

1. Найти медиану массива с помощью процедуры рандомизированного выбора RSelect
2. Найти медиану массива с помощью процедуры детерминированного выбора DSelect

Использовать два тестовых примера, затем сформировать массив из первых цифр в десятичной записи числа пи (см. ниже). Сравнить временные затраты для задачи 1 и для задачи 2.

**Тестовый пример № 1:** этот файл содержит 10 целых чисел, представляющих массив из 10 элементов. Найти медиану (то есть 5-й наименьший элемент)?

**Тестовый пример № 2:** этот файл содержит 100 целых чисел, представляющих массив из 100 элементов. Найти медиану (т. е. статистику 50-го порядка)?

**Набор данных задачи:** сформируйте массив из 1000 элементов, в котором первый элемент — это первые 10 цифр в десятичной записи числа пи, второй элемент — следующие 10 цифр числа пи и так далее. (Цифры числа пи доступны здесь .) Что такое медиана массива? Есть ли в этом массиве повторяющиеся элементы?

Сравнить временные затраты для задачи 1 и для задачи 2.

Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 8. Вычисление сильно связанных компонентов

**Тестовый пример № 1:** граф с 9 вершинами и 11 ребрами. Топ 5 размеров SCC: 3,3,3,0,0

**Тестовый пример № 2:** граф с 8 вершинами и 14 ребрами. Топ 5 размеров SCC: 3,3,2,0,0

**Тестовый пример № 3:** граф с 8 вершинами и 9 ребрами. Топ 5 размеров SCC: 3,3,1,1,0

**Тестовый пример № 4:** граф с 8 вершинами и 11 ребрами. Топ 5 размеров SCC: 7,1,0,0,0

**Тестовый пример № 5:** граф с 12 вершинами и 20 ребрами. Топ 5 размеров SCC: 6,3,2,1,0

**Набор данных задачи:** этот файл описывает ребра ориентированного графа. Вершины обозначаются целыми положительными числами от 1 до 875714. Каждая строка обозначает одно ребро графа (хвостовую и головную вершины в указанном порядке). Например, одиннадцатая

строка («2 13019») указывает на то, что существует ребро, направленное из вершины 2 в вершину 13019. Каковы размеры пяти самых больших компонент сильной связности?

Каково время выполнения задачи?

Привести и обосновать асимптотическую оценку временной сложности

## 9. Алгоритм Дейкстры

1. Составьте программу, осуществляющую простую реализацию алгоритма Дейкстры. Используйте ее для решения задачи о кратчайшем пути с единтвенным источником в разных неориентированных графах

2. Составьте программу, реализующую кучевую версию алгоритма Дейкстры. Используйте ее для решения задачи о кратчайшем пути с единственным источником в разных неориентированных графах

**Тестовый пример:** этот файл описывает неориентированный граф с 8 вершинами (формат файла см. ниже). Каковы кратчайшие расстояния от вершины 1 до любой другой вершины? (Ответ, для вершин с 1 по 8, в порядке: 0,1,2,3,4,4,3,2.)

**Набор данных задачи:** этот файл содержит представление списка смежности неориентированного графа с 200 вершинами, помеченными от 1 до 200. В каждой строке указаны ребра, инцидентные данной вершине, а также их (неотрицательные) длины. Например, в шестой строке первой записью является «6», указывающая, что эта строка соответствует вершине 6. Следующая запись в этой строке «141,8200» указывает, что между вершиной 6 и вершиной 141 существует ненаправленное ребро, имеющее длина 8200. Остальные пары в этой строке указывают остальные вершины, смежные с вершиной 6, и длины соответствующих ребер. Вершина 1 является начальной вершиной. Каковы кратчайшие расстояния от вершины 1 до следующих десяти вершин? 7,37,59,82,99,115,133,165,188,197.

Сравнить временные затраты для задачи 1 и для задачи 2.
Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 10. Поддержка медианы

Составить программу, реализующую алгоритм поддержки медианы с помощью двух бинарных куч.

**Тестовый пример:** этот файл представляет собой поток из 10 чисел. Каковы последние 4 цифры суммы k-х медиан? (См. ниже определение k-й медианы.)

**Набор данных испытания:** этот файл содержит список целых чисел от 1 до 10000 в несортированном порядке; вы должны относиться к этому как к потоку чисел, поступающих одно за другим. Под k-й медианой понимается медиана первых k чисел в потоке

(((k+1)/2)-е наименьшее число среди первых k, если k нечётное, (k/2)-е наименьшее, если k четное).

Каковы последние 4 цифры суммы k-х медиан (при k от 1 до 10000)? Какая структура данных делает ваш алгоритм быстрее: две кучи или дерево поиска?

Каково время выполнения задачи?

Привести и обосновать асимптотическую оценку временной сложности.

## 11. Реализовать хеш-табличное решение задачи 2-СУММ (о сумме двух чисел)

(определить, существуют ли среди элементов числового массива два числа, сумма которых равна заданному числу)

1. Составить программу о сумме двух чисел на основе отсортированного массива.
2. Составить программу о сумме двух чисел на основе хеш-таблиц.

**Тестовый пример:** этот файл описывает массив из 9 целых чисел. Для скольких целевых значений t в интервале [3,10] существуют различные числа x,y во входном массиве такие, что x+y=t?

**Набор данных испытания:** этот файл содержит миллион целых чисел, как положительных, так и отрицательных (возможно, с повторениями!), где i-я строка определяет i-ю запись входного массива. Для скольких целевых значений t в интервале [-10000,10000] существуют различные числа x,y во входном массиве, такие что x+y=t?

Сравнить временные затраты для задачи 1 и для задачи 2.
Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 12. Жадное планирование. Минимизация взвешенной суммы сроков завершения работ.

1. Реализовать алгоритм GreedyDiff
2. Реализовать алгоритм GreedyRatio

**Тестовый пример:** (предоставлен Джереми Брауном). Этот файл содержит список из 12 заданий с весами и длинами. Он имеет формат:

[число_заданий] [вес_задания_1] [длина_задания_1]
[вес_задания_2] [длина_задания_2]
...

Какова взвешенная сумма времени выполнения расписания, выводимого алгоритмами GreedyDiff и GreedyRatio? (Разорвите ничьи в пользу работ с большими весами.)

**Набор данных задачи:** повторите предыдущую задачу с набором из 10 000 заданий, перечисленных в этом файле .

Сравнить временные затраты для задачи 1 и для задачи 2.
Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

## 13. Минимальные остовные деревья

1. Составить алгоритм Прима на основе двоичной кучи.
2. Составить алгоритм Кускала на основе структуры Union_Find

В этой задаче формат файла:

[число_вершин] [число_ребер] [одна_конечная_точка_ребра_1]
[другая_конечная_точка_ребра_1] [стоимость_ребра_1] [одна_конечная_точка_ребра_2]
[другая_конечная_точка_ребра_2] [стоимость_ребра_2]

...
Стоимость ребер может быть отрицательной и не обязательно различна.
**Тестовый пример:** (предоставлено Квентином Эпплби) Какова стоимость MST на графа, описанном в этом файле? (Ответ: 14)
**Набор данных задачи:** повторите предыдущую задачу для графа, описанного в этом файле .

Сравнить временные затраты для задачи 1 и для задачи 2.
Привести и обосновать асимптотическую оценку временной сложности обоих алгоритмов.

14. Взвешенное независимое множество. Найти независимое множество вершин в путевом графе, имеющее максимальной вес.
1. Составить программу, вычисляющую максимальный суммарный вес независимого множества (WIS).
2. Составить программу, которая строит независимое множество вершин максимального суммарного веса (WIS_RECONSTRUCTION).
В этой задаче каждый файл описывает веса вершин в графе путей и имеет формат:
[число_вершин_в_графе_путей]
[вес первой вершины]
[вес второй вершины]
...
**Тестовый пример:** (предоставлено Логаном Трэвисом) Что такое значение независимого множества максимального веса 10-вершинного графа путей, описанного в этом файле , и какие вершины принадлежат MWIS?
**Набор данных задачи:** повторите предыдущую задачу для графа путей из 1000 вершин, описанного в этом файле .
Каково время выполнения задачи?

Привести и обосновать асимптотическую оценку временной сложности.

15. Задача о рюкзаке
Найти множество предметов с максимально возможной суммой значений, при условии, что суммарный размер предметов не превосходит заданного C.
1. Составить программу, вычисляющую максимальную сумму значений предметов (Knapsack)
2. Составить программу, которая находит множество предметов с максимальной суммой значений (Knapsack_Reconstruction).

В этой задаче каждый файл описывает экземпляр задачи о рюкзаке и имеет формат:
[ размер_рюкзака][
количество_предметов] [значение_1] [вес_1]
[значение_2] [вес_2]

...
Можно считать, что все числа положительные. Вы должны исходить из того, что вес предмета и вместимость рюкзака являются целыми числами.
**Тестовый пример:** какова ценность оптимального решения для экземпляра ранца, описанного в этом файле ?
***Набор данных задачи:** повторите предыдущую задачу для экземпляра ранца, описанного в этом файле . Этот экземпляр настолько велик, что простая итеративная реализация, описанная в книге, требует неосуществимого количества времени и пространства. Поэтому вам придется проявить творческий подход, чтобы найти оптимальное решение. Одна из идей состоит в том, чтобы вернуться к рекурсивной реализации, решая подзадачи --- и, конечно же, кэшируя результаты, чтобы избежать избыточной работы --- только по мере необходимости. Кроме того, не забудьте подумать о подходящих структурах данных для хранения и поиска решений подзадач.

Каково время выполнения задачи?

Привести и обосновать асимптотическую оценку временной сложности.

## 16. Выравнивание последовательностей

1. Составить программу, вычисляющую NW- отметку символьных цепочек X и Y.
2. Составить программу, которая реконструирует последовательности X иY по найденной NW- отметке.

Этот файл описывает пример задачу выравнивания последовательности. Формат файла:
1-я строка: длина X и длина Y
2-я строка: стоимость пробела и стоимость несоответствия (последняя одинакова для каждой пары различных символов)
3-я строка: последовательность X
4-я строка: последовательность Y