МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Алгоритм Грехема и сортировка 5, 10 кучей»**

**Выполнил:**

Cтудент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Доцент кафедры АГДМ

Малышев Д. С.

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2019

**Оглавление**

[1.Введение 3](#_Toc27170422)

[2.Постановка задачи 4](#_Toc27170423)

[3.Руководство пользователя 5](#_Toc27170424)

[4.Руководство программиста 6](#_Toc27170425)

[5.Эффективность 7](#_Toc27170426)

[6.Заключение 8](#_Toc27170427)

[7.Список литературы 9](#_Toc27170428)

# 1.Введение

В данной лабораторной работе рассматриваются алгоритмы построения выпуклой оболочки Грехема и сортировка 5 и 10 кучей. Алгоритм Грехема – это алгоритм построения выпуклой оболочки в двумерном пространстве. Выпуклая оболочкой называется наименьшее выпуклое множество содержащие полное множество точек. Сортировка 5, 10 кучей представляет из себя построение приоритетной очереди на куче с последовательным удалением элементов из начала очереди и последующем применением операции погружение.

# 2.Постановка задачи

Целью данной работы является разработка эффективного алгоритма:

1. D heap sort
2. Convex hull

# 3.Руководство пользователя

При запуске программы для построения выпуклой оболочки создается 5 случайных векторов из точек длинной 100000, 200000, 300000, 400000 и 500000. Применяется нахождение выпуклой оболочки и считается время.

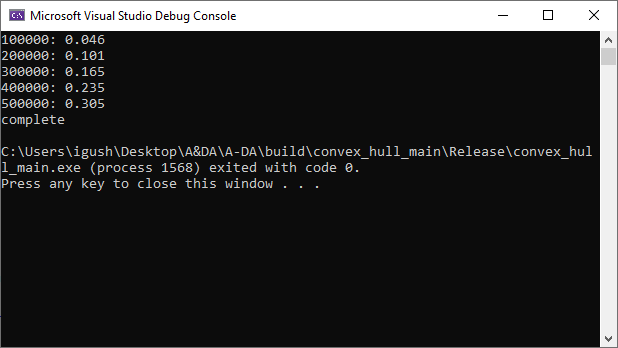


Рис1. Пример работы программы.

При запуске программы для сортировки массива создается 5 случайных векторов из int длинной 100000, 200000, 300000, 400000 и 500000. Применяется 5 и 10 d heap sort и считается время.

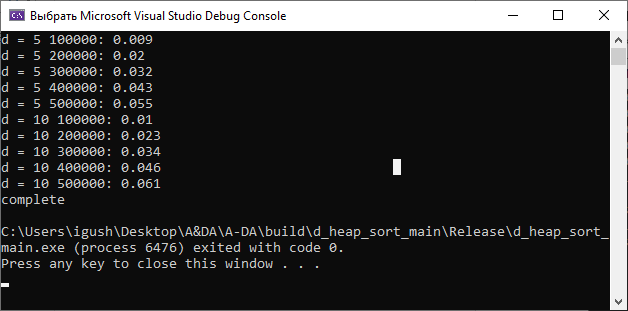


Рис1. Пример работы программы.

# 4.Руководство программиста

Для Convex hull

1. void setAngle(std::vector<Point>& A) - считает длину и угол от начальной точки.
2. void setFirstPoint(std::vector<Point>& vec) – находит начальную точку.
3. std::vector<Point> createHull(const std::vector<Point>& vec) – создает выпуклую оболочку.

Для D\_heap\_sort

1. void dHeapSort(std::vector<T>& vec, int d) – сортирует массив.
2. void dive(std::vector<T>& vec, int d, int i, int size) – опускает элемент по дереву в зависимости от его веса.
3. int minChild(std::vector<T>& vec, int d, int i, int size) – ищет потомка с минимальным весом.

# 5.Эффективность

График для функции dHeapSort



Можно заметить, что вариант на 5 потомках имеет лучшее время на всем промежутке.

Сложность алгоритма O() где d = количеству потомков.

График для Convex hull



Сложность зависит от применяемой сортировки. В данном случае сложность составляет O().

# 6.Заключение

Благодаря данной работе я изучил и разработал метод сортировки на куче и построение выпуклой оболочки, а также стал лучше понимать оценку сложности алгоритмов.

# 7.Список литературы

В.Е. Алексеев, В.А. Таланов Графы. Модели вычислений. Структуры данных.