**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЁТ**

**по лабораторной работе №2**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Решение задачи принципом «Разделяй и властвуй»**

Студент гр. 0322 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руссу В.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пестерев Д.О.

Санкт-Петербург

2022

# Задача

О влюбленных улитках. На квадрате земли 1 км2 располагаются улитки-гермафродиты. В момент времени каждая из улиток с постоянной скоростью 1 cм/с ползет к улитке, являющейся ближайшей к ней в момент времени t = 0, выбрав её в качестве спутника жизни. Определить время, через которое первая пара улиток достигнет друг друга или наличие ситуации, приводящей улиток в замешательство, наиболее эффективным способом

# Описание реализуемого класса и методов

Структура Snail содержит в себе поля с координатами улиток, массив со всеми возможными путями и метод, который ищет не находится ли улитка в замешательстве.

Функции mergeSort и mergе отвечают за сортировку массива с использованием принципа “Разделяй и властвуй”. Функция pathlength возвращает расстояние между двумя точками.

void if\_confusion () сортирует массив расстояний и проверяет будет ли улитка в замешательстве. Временная сложность: О(n\*log(n)).

double pathlength(double x1, double y1, double x2, double y2) возвращает расстояние между двумя улитками. Временная сложность: О(1).

void mergeSort(double\* arr, int l, int r) Временная сложность: О(n\*log(n)).

void Merge(double\* array, int start, int mid, int end) Временная сложность: О(n)

**Теоретически асимптотическая сложность реализованного алгоритма**: O(n\*log(n)).

**Оценка временной сложности эмпирическим способом:**

Слева количество элементов – справа время выполнения метода разделяй и властвуй.

2 элемента - 0.00010 миллисекунд

4 элемента - 0.00195 миллисекунд

8 элемента - 00409 миллисекунд

12 элемента - 0.00660 миллисекунд

16 элемента - 0.00857 миллисекунд

20 элемента - 0.01150 миллисекунд

32 элемента - 0.01833 миллисекунд

53 элемента - 0.03194 миллисекунд

64 элемента - 0.03893 миллисекунд

128 элемента - 0.08109 миллисекунд

График подобает теоретическо-асимптотической сложности из пункта 3 (Рисунок 1).

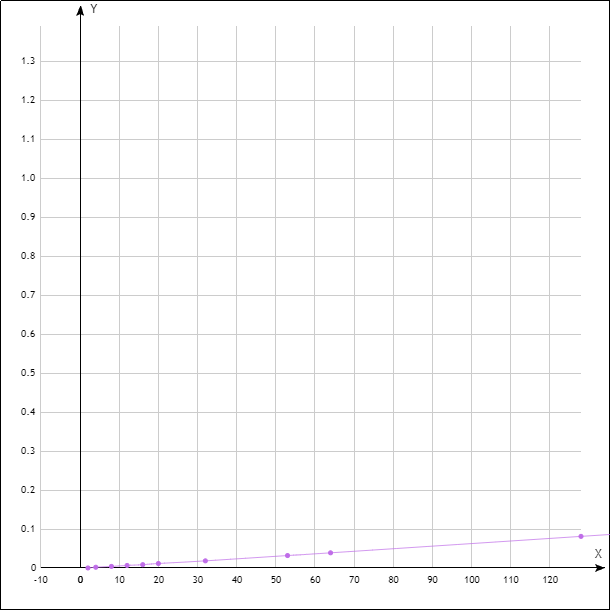


Рисунок 1

# Пример работы

Вводим координаты четырех улиток: (0, 0), (100, 0), (0, 100), (15, 100)

Ожидаемый результат: Улитка под номером 0 в замешательстве, так как ближайшие улитки находятся на одном расстоянии от нее. Кратчайший время равно 15.

Результат: Рисунок 2.

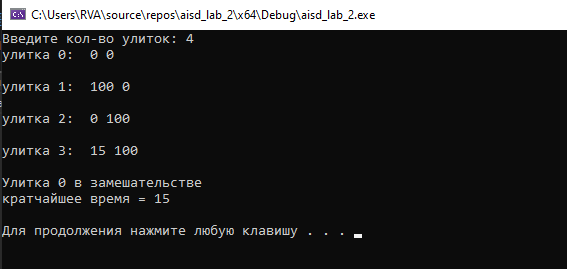


Рисунок 2

**Ссылка на репозиторий**

https://github.com/RVA-t/aisd\_lab\_2.git