# Отчёт

Тема: Визуализация растровых алгоритмов (Пошаговый, ЦДА, Брезенхем — отрезок и окружность)

Ф.И.О.: Богуш Полина Рихардовна

Курс: 2

Группа: 1

## 1. Цель

Реализовать и сравнить по скорости работы базовые растровые алгоритмы построения отрезков и окружностей.

## 2. Реализация

Python 3 + tkinter. каждая клетка — пиксель. Алгоритмы возвращают список координат пикселей, которые затем визуализируются.

## 3. Измерения времени

Замеры выполнялись через time.perf\_counter() (800 повторов, медиана). Отрисовка не учитывалась.

|  |  |
| --- | --- |
| Алгоритм / случай | median (µs) |
| Пошаговый — короткий | 4.32 |
| ЦДА — короткий | 4.28 |
| Брезенхем (отрезок) | 1.60 |
| Пошаговый — средний | 13.09 |
| ЦДА — средний | 13.07 |
| Брезенхем — средний | 5.38 |
| Пошаговый — длинный | 20.59 |
| ЦДА — длинный | 20.40 |
| Брезенхем — длинный | 8.93 |
| Окружность r=5 | 4.72 |
| Окружность r=20 | 15.31 |
| Окружность r=35 | 26.01 |

## 4. Вывод

Алгоритм Брезенхема выполняется быстрее (~в 2–3 раза), так как использует только целые операции. Рост времени для окружности пропорционален радиусу (O(r)).

## 5. Пример измерений (bench.py)

import time, statistics  
from algorithms import step\_coords, dda\_coords, bresenham\_line\_coords, bresenham\_circle\_coords  
  
def bench(func, \*args, runs=800):  
 times = []  
 for \_ in range(runs):  
 t0 = time.perf\_counter()  
 func(\*args)  
 t1 = time.perf\_counter()  
 times.append(t1 - t0)  
 return statistics.median(times)  
  
print("step:", bench(step\_coords, 10, 10, 30, 15))  
print("dda:", bench(dda\_coords, 10, 10, 30, 15))  
print("bres line:", bench(bresenham\_line\_coords, 10, 10, 30, 15))  
print("bres circle:", bench(bresenham\_circle\_coords, 50, 50, 20))