**ПРАКТИЧЕСКАЯ РАБОТА № 2**

**«Оценка сложности алгоритмов поиска»**

1. Запустили проект на выполнение, убедились в правильности работы программы:

import random

import time

def get\_sort\_list(n: int) -> list:

    return [i for i in range(n)]

def find(l: list, key: int) -> int:

    bp = 0

    ep = len(l) - 1

    mp = -1

    while bp <= ep:

        mp = bp + (ep - bp) // 2

        time.sleep(0.001)

        if l[mp] == key:

            return mp

        if l[mp] < key:

            bp = mp + 1

        else:

            ep = mp - 1

    return mp

def main():

    for j in range(1, 10):

        i = j \* 5

        l = get\_sort\_list(i)

        print(\*l)

        k = random.randint(0, i)

        t0 = time.time()

        p = find(l, k)

        t1 = time.time()

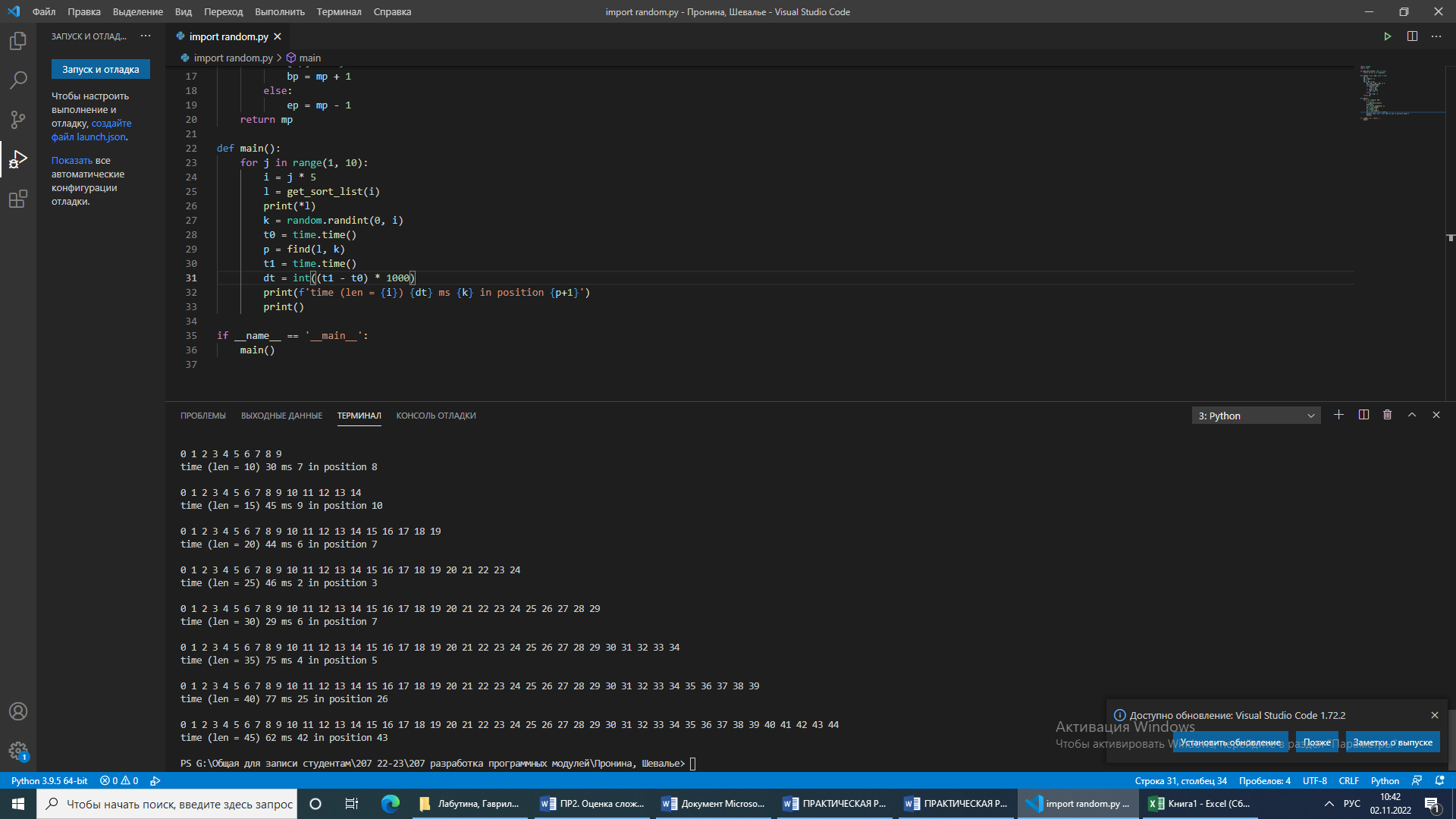
        dt = int((t1 - t0) \* 1000)

        print(f'time (len = {i}) {dt} ms {k} in position {p+1}')

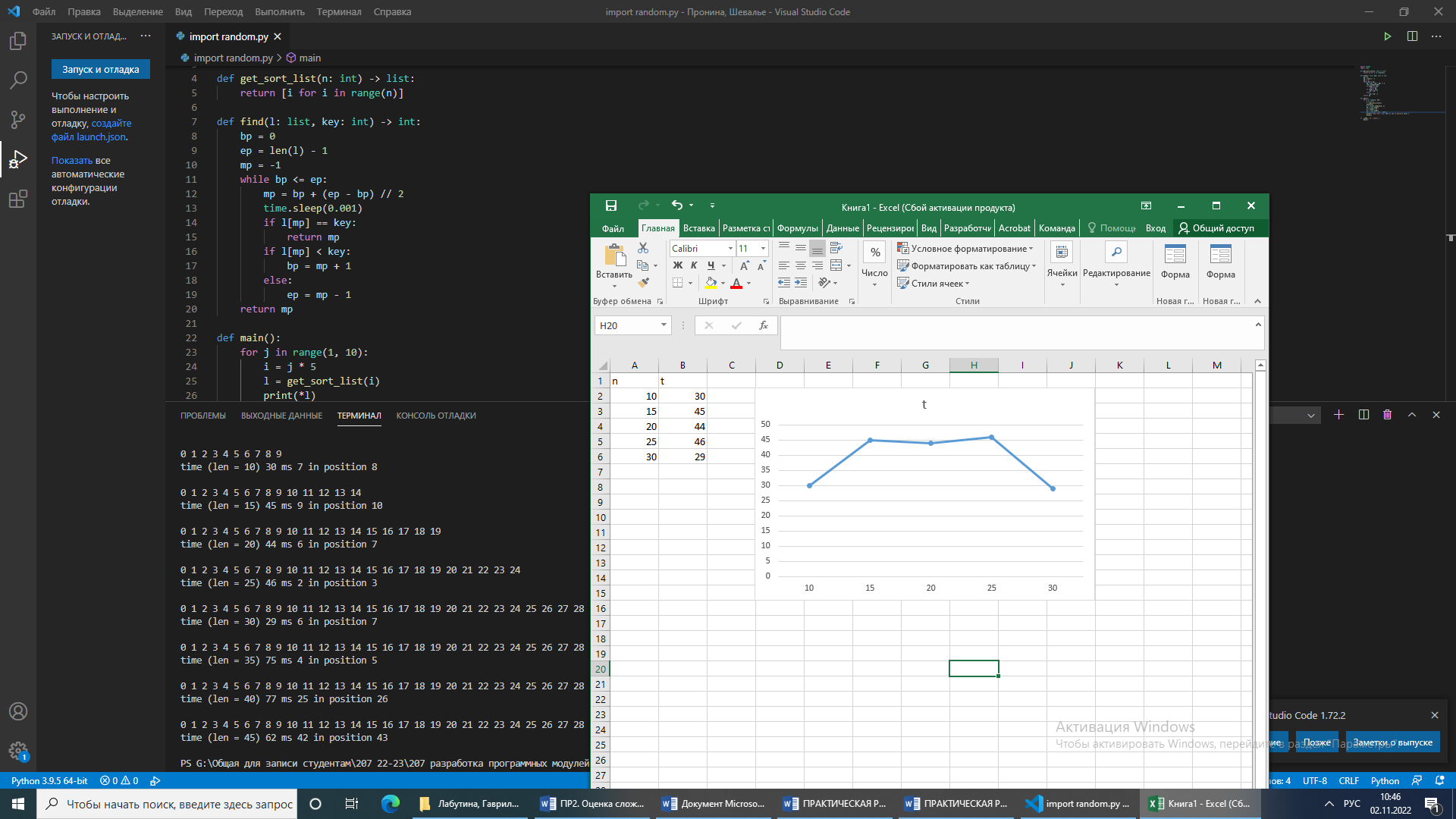
        print()

if \_\_name\_\_ == '\_\_main\_\_':

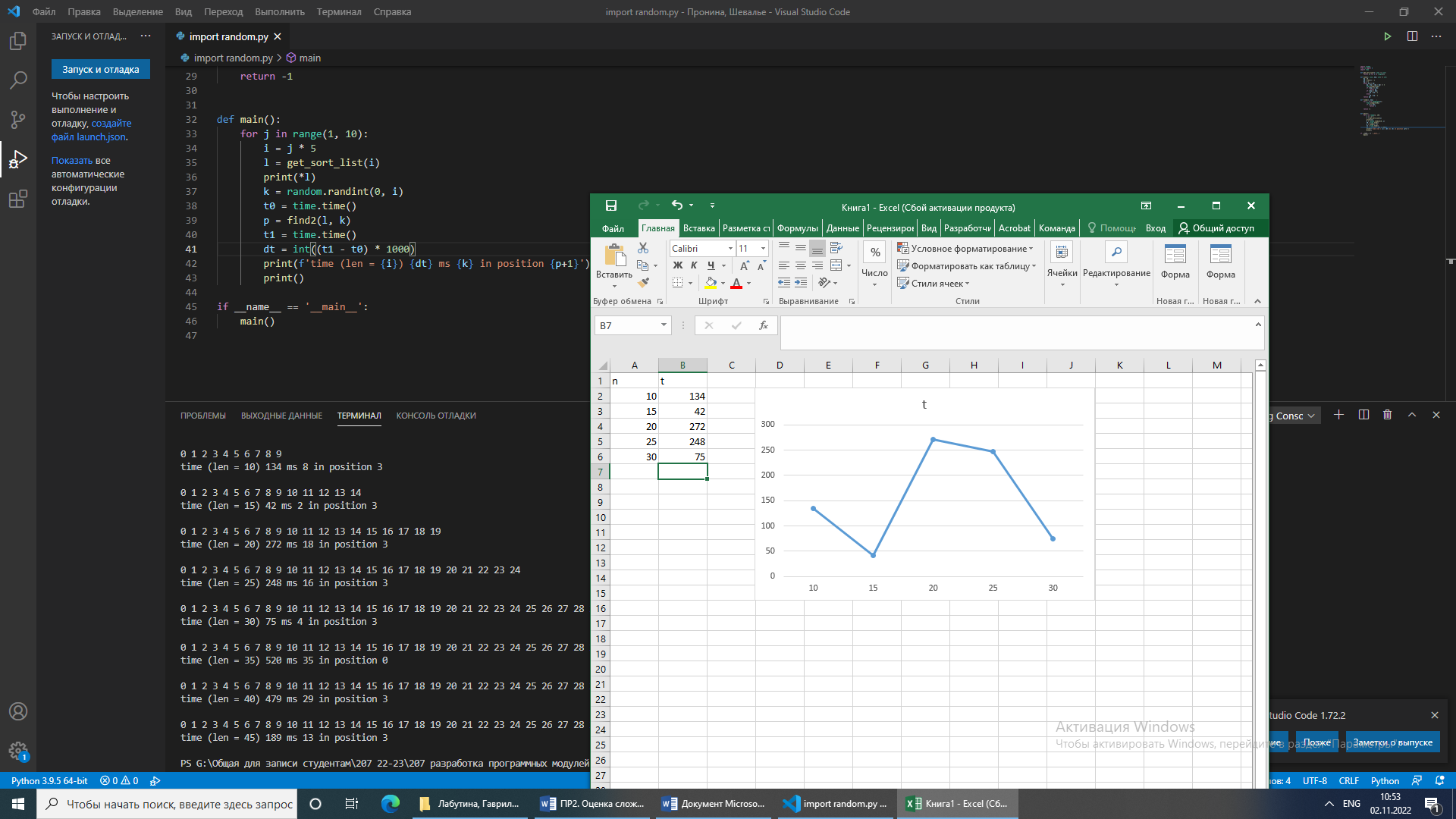
    main()



**2.** Записали результаты выполнения в таблицу Excel и построили график зависимости времени выполнения от кол-ва массива:



**3.** Изменили алгоритм поиска, внесли изменения в программу и записали результаты выполнения в таблицу Excel, а также построили график зависимости времени выполнения от кол-ва массива:



**Вывод:** основываясь на первом графике, мы смогли определить, что зависимость является не прямой, а возрастает по формуле **log2(n).** Основываясь на втором графике, мы смогли определить, что зависимость является прямой.