# PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

**Modul 10**

**Analisi Algoritma**



# Disusun oleh:

# DONI WAHYU SAPUTRO

# L200200169

**G**

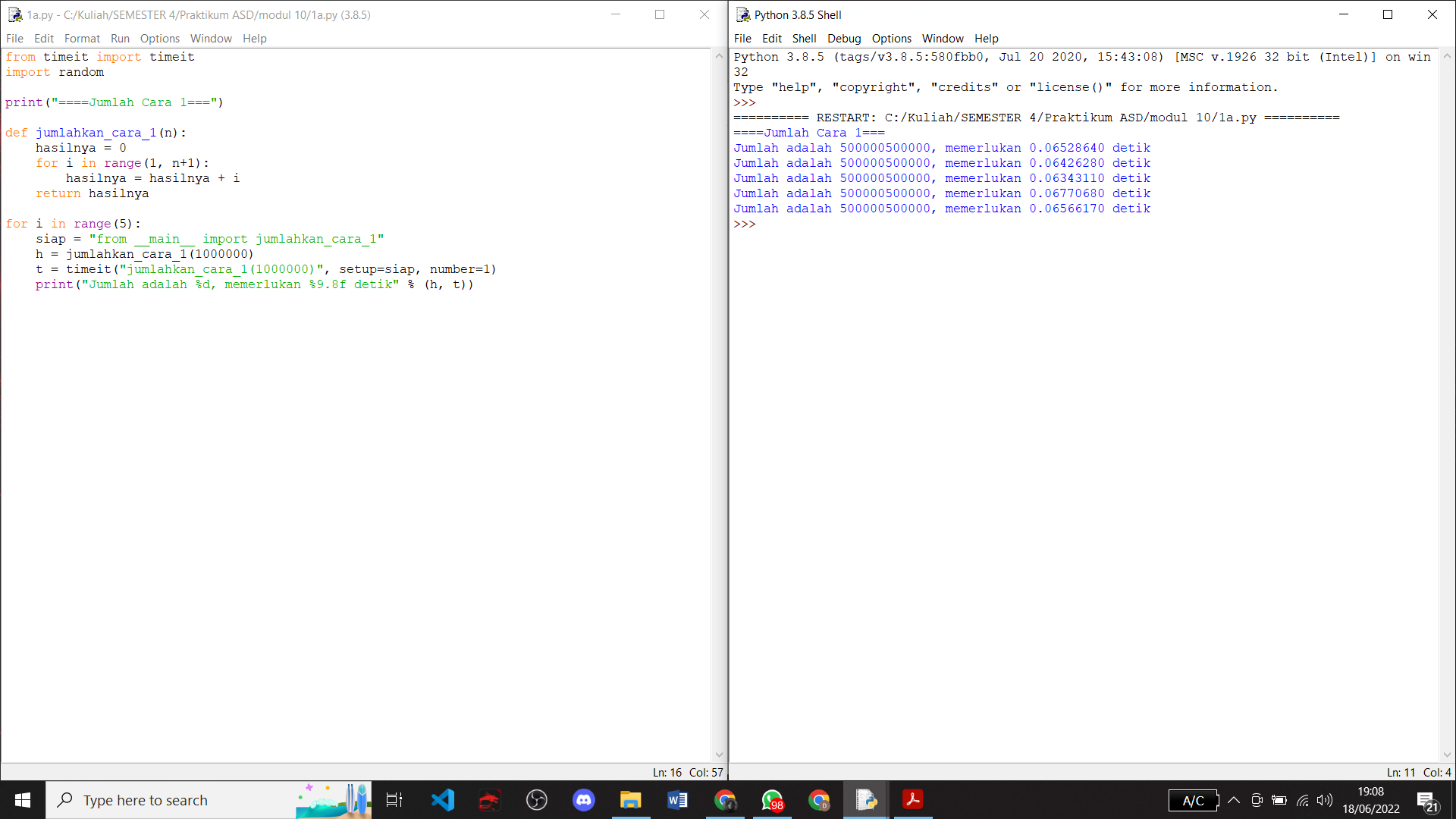
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

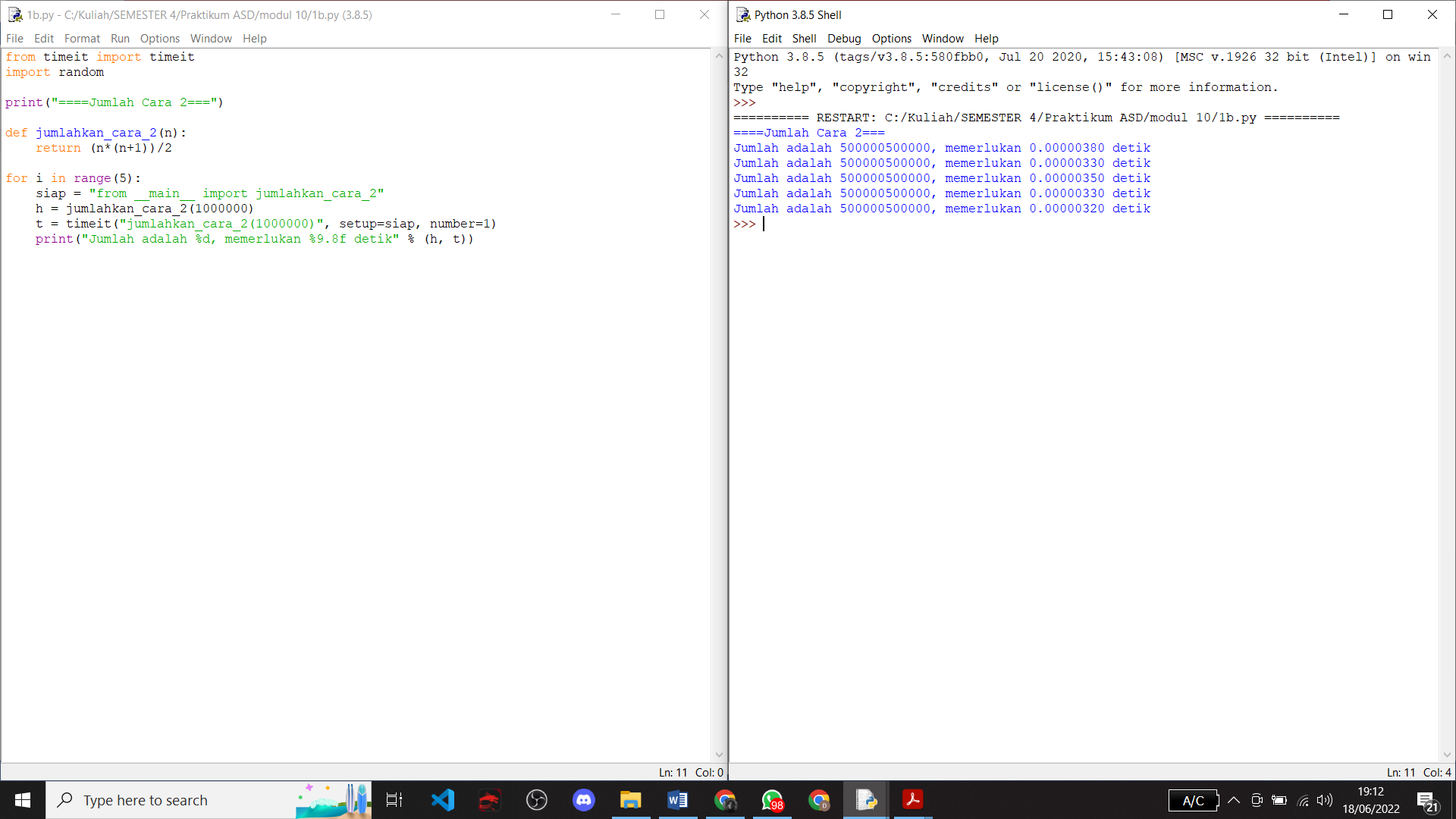
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

Latihan

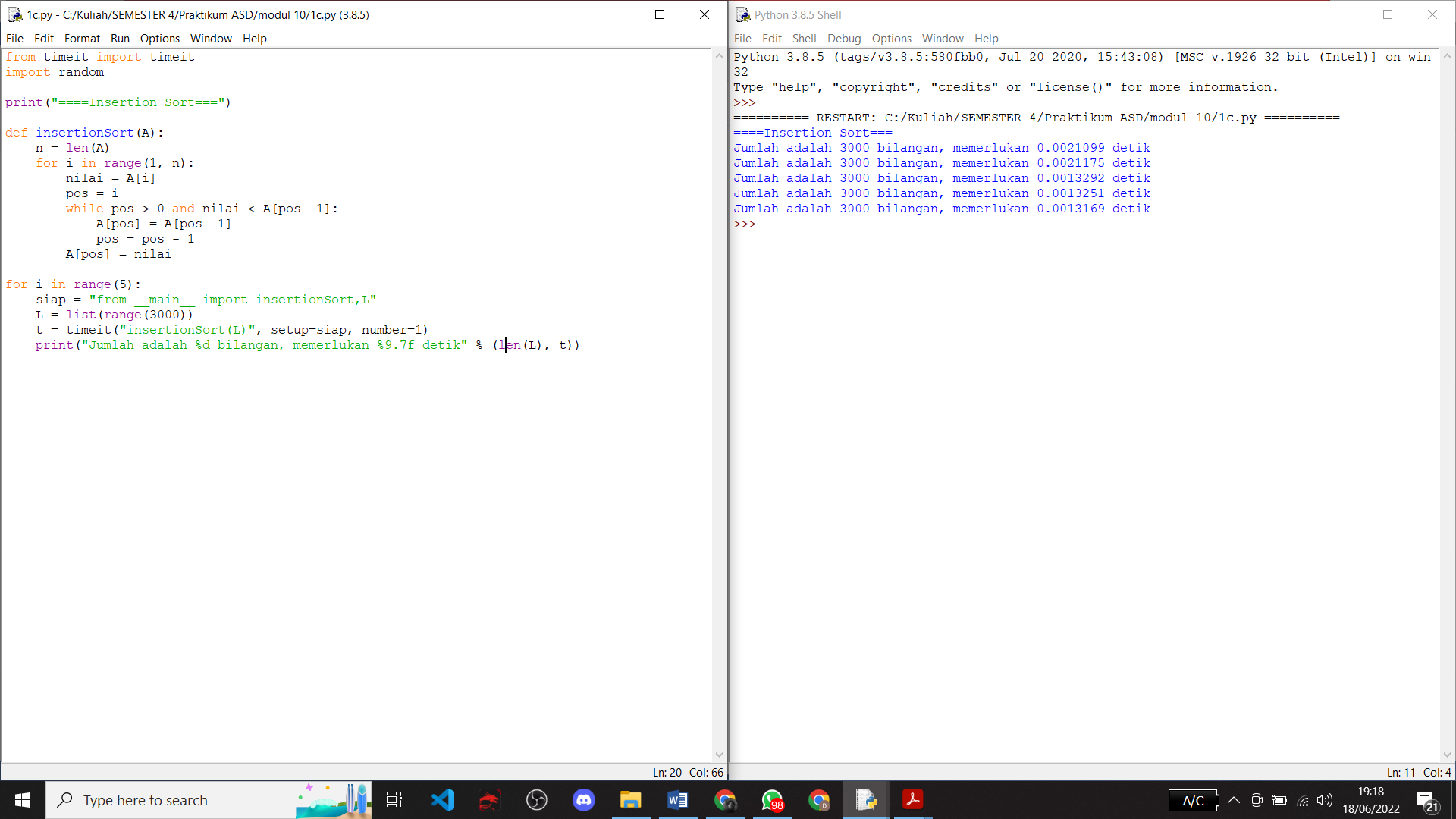
1.

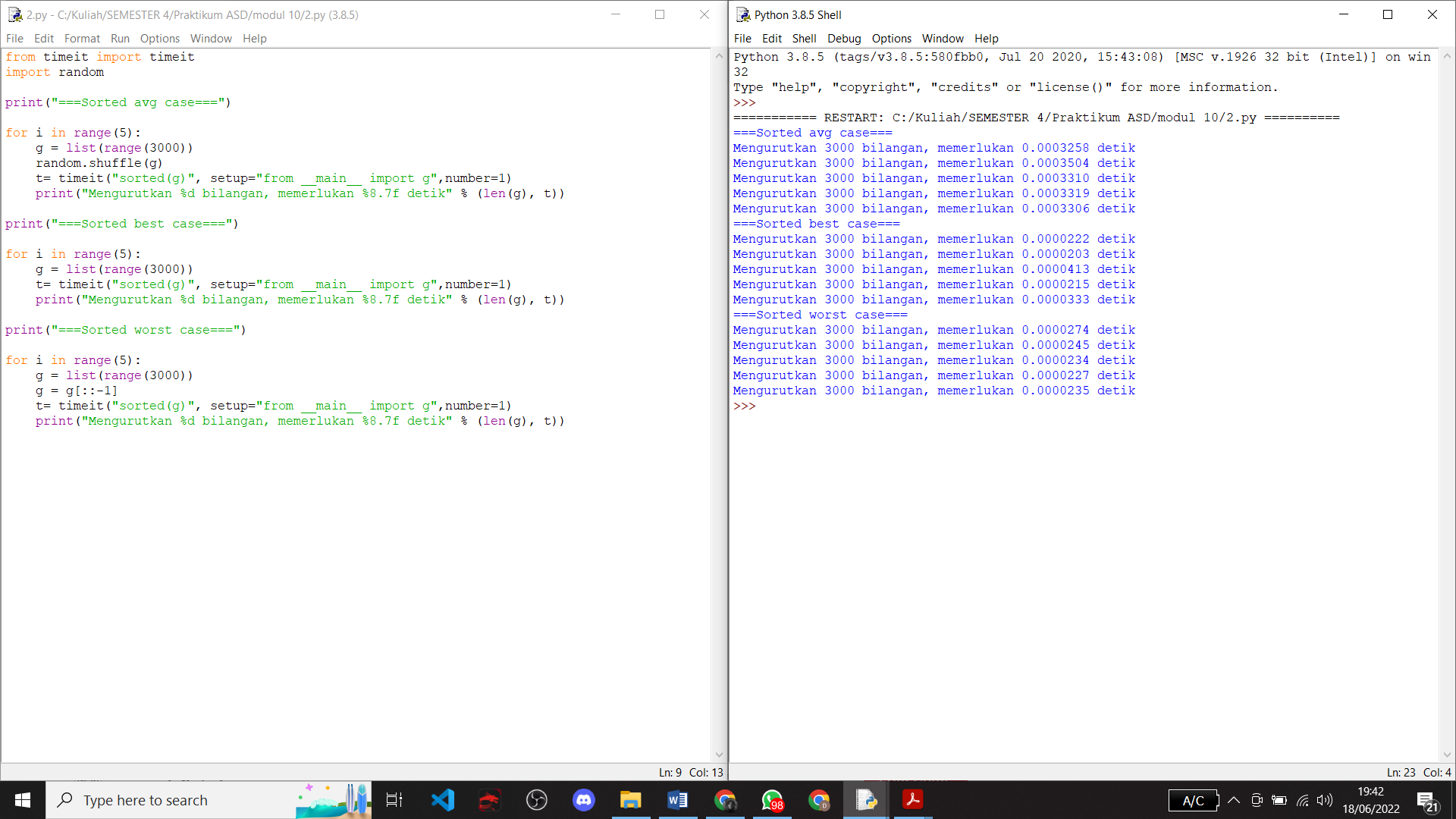
a 

b



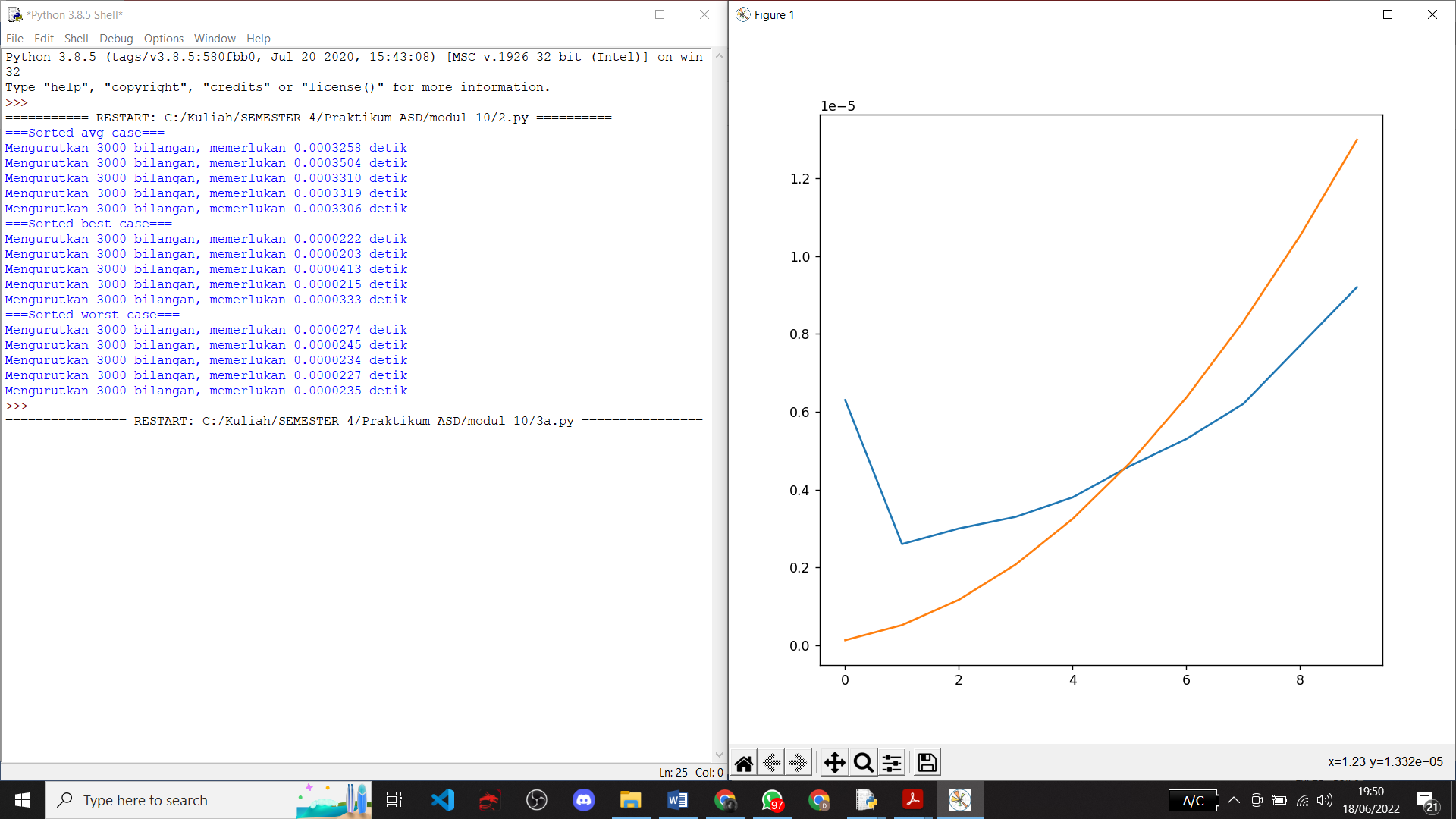
c



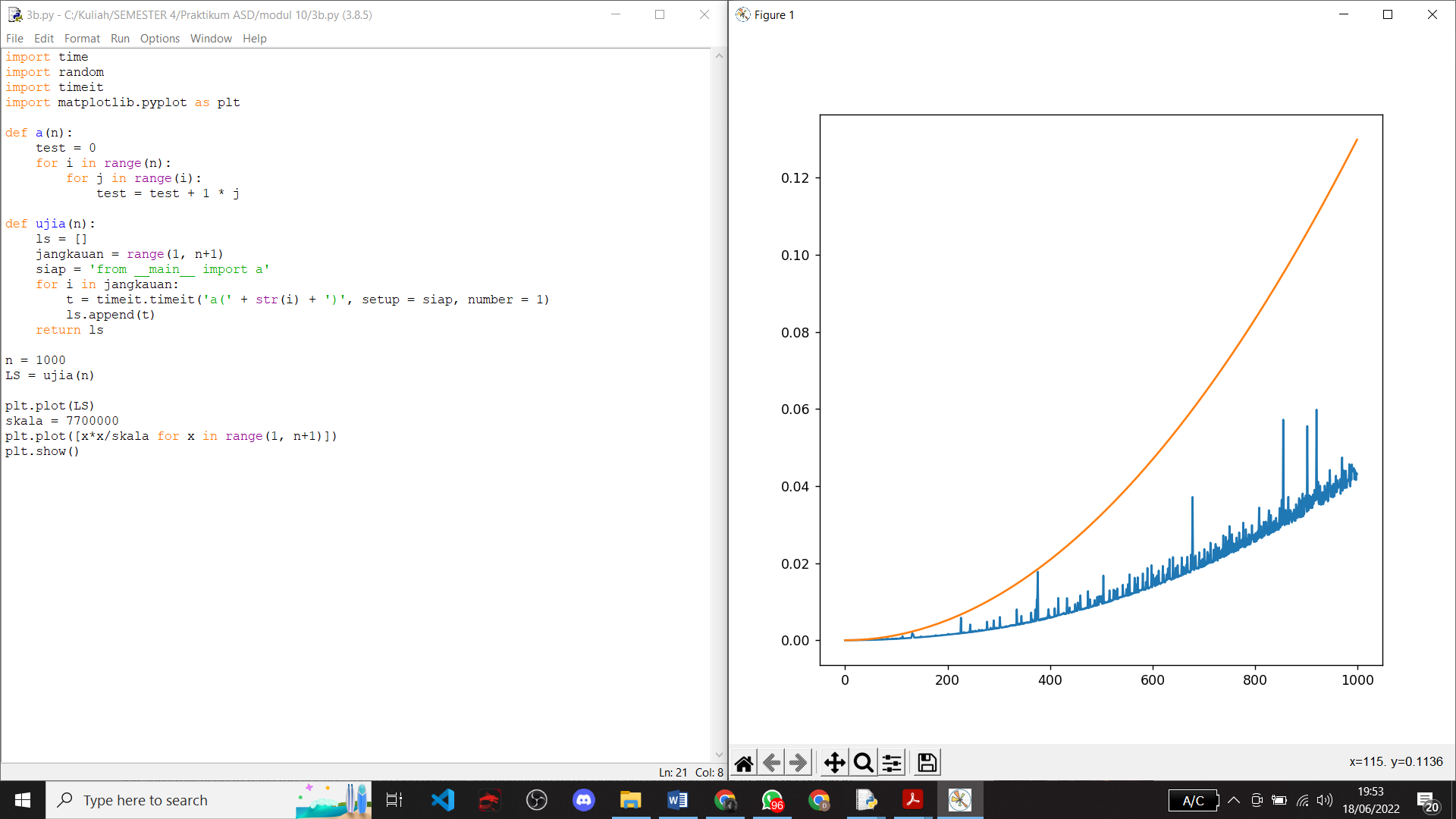
2. 

3.

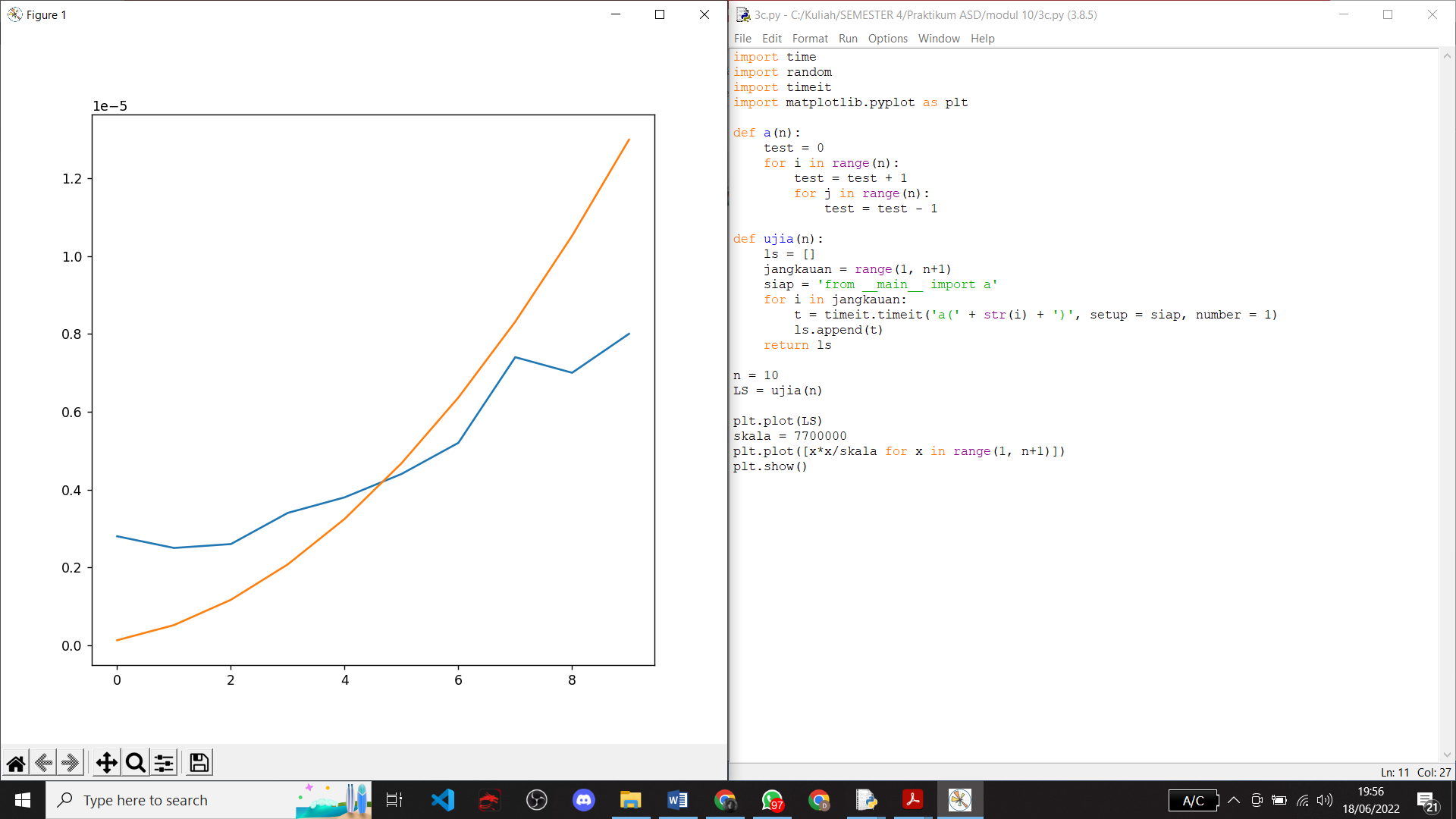
a



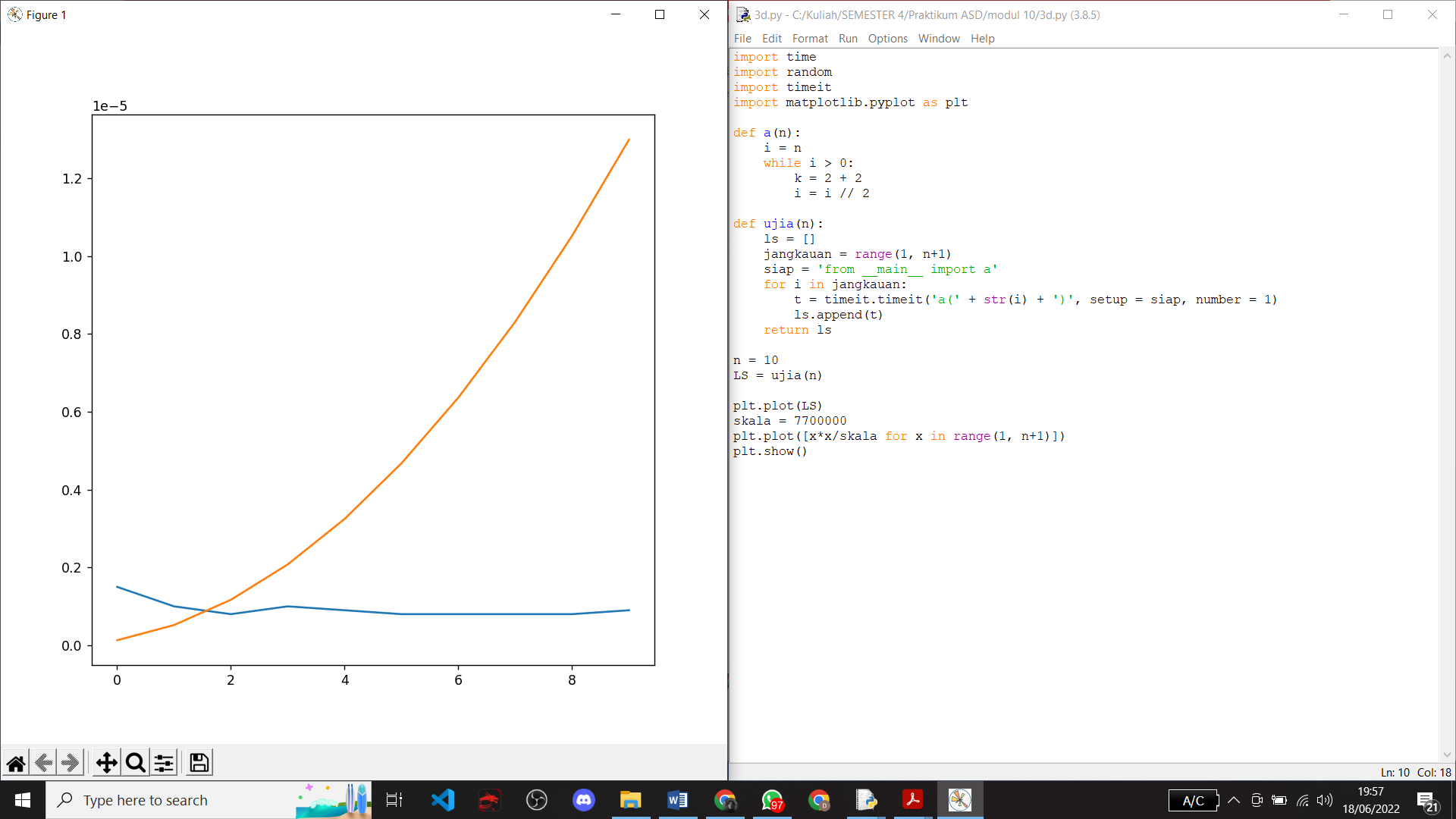
b



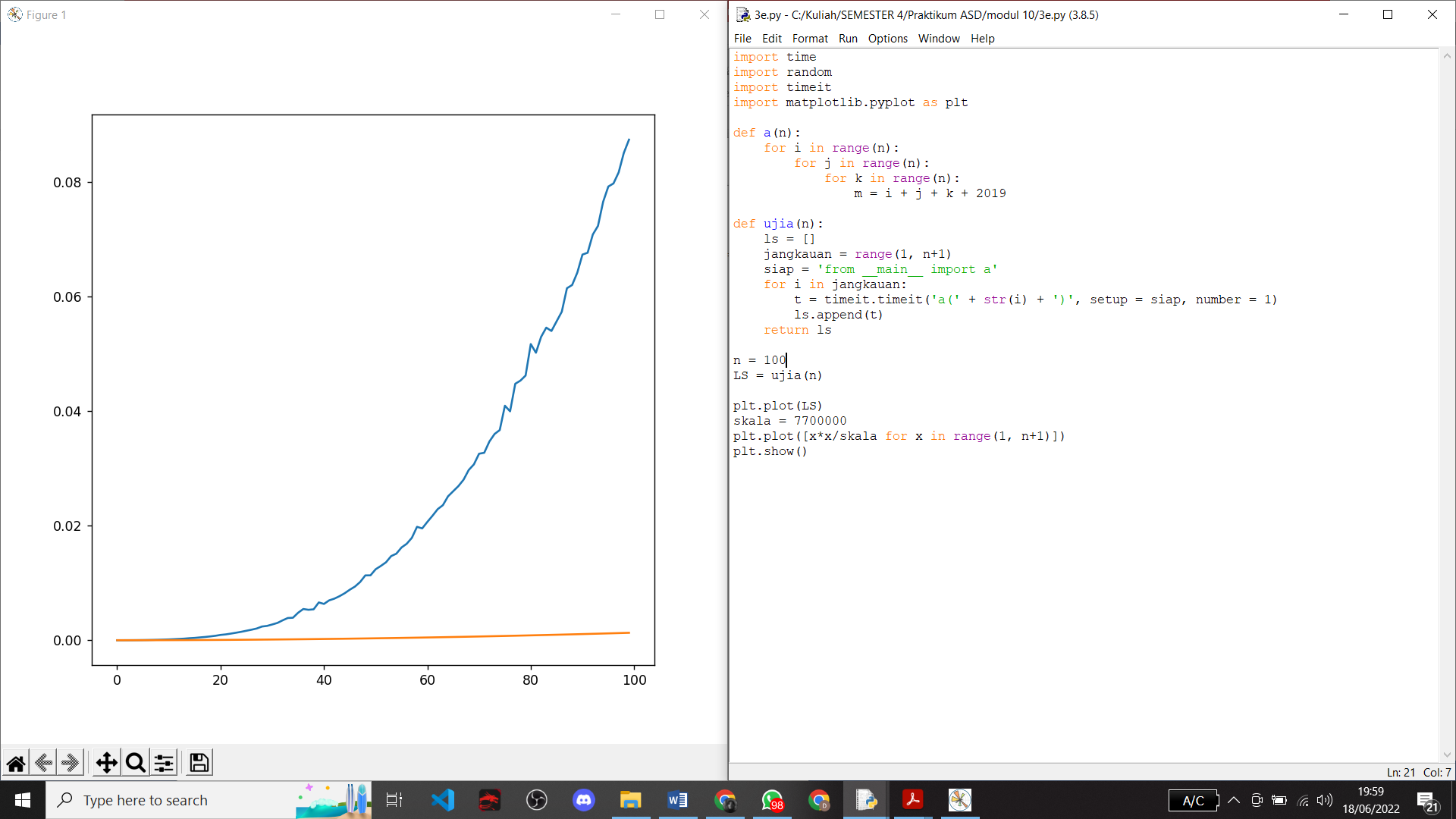
c



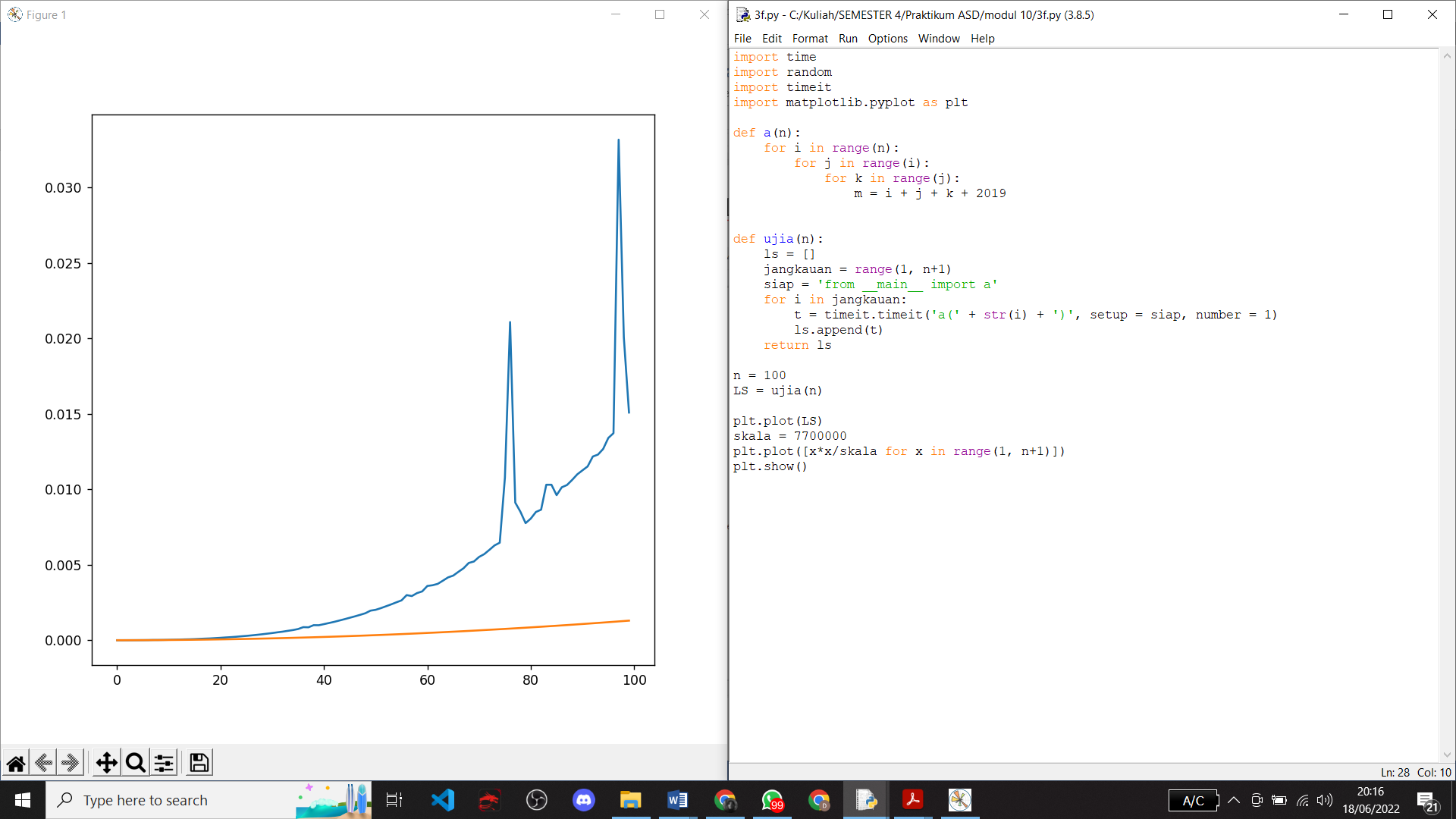
d



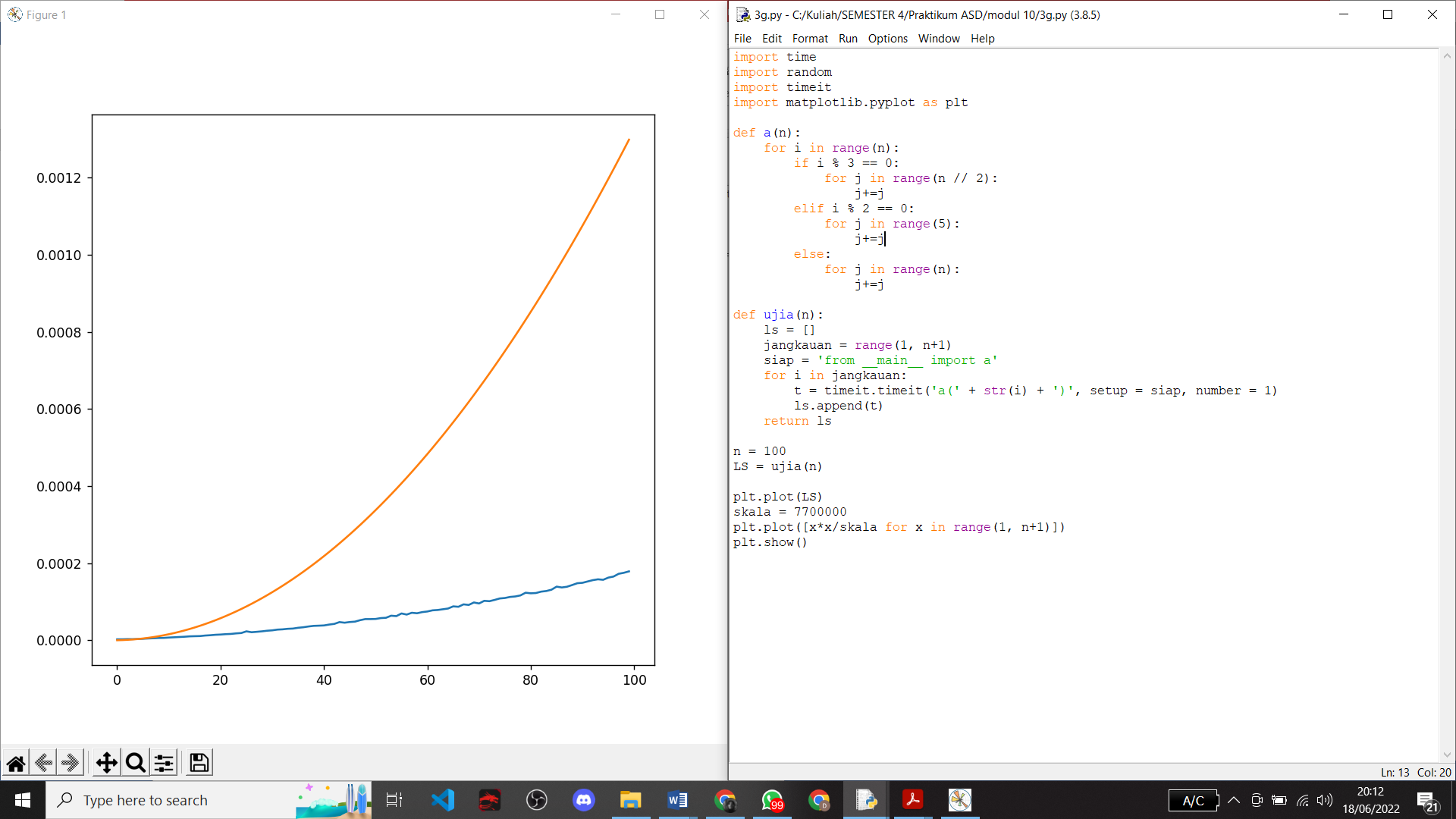
e



f



g



4. Urutkan dari yang pertumbuhan kompleksitasnya lambat ke yang cepat log4n < 10log2n < n log2n < 2 log2n < 5n2 < n3 < 12n6 < 4n

5. Tentukan O(.) dari fungsi-fungsi berikut, yang mewakili banyaknya langkah yang diperlukan untuk beberapa algoritma

a. *T*(*n*) = *n*2 + 32*n* + 8 = O(n^2)

b. *T*(*n*) = 87*n* + 8*n* = O(n)

c. *T*(*n*) = 4*n* + 5*n* log *n* + 102 = O(n log n)

d. *T*(*n*) = log *n* + 3*n*2 + 88 = O(n^2)

e. *T*(*n*) = 3(2*n*) + *n*2 + 647 = O(2^n)

f. *T*(*n, k*) = *kn* + log *k =* O(kn)

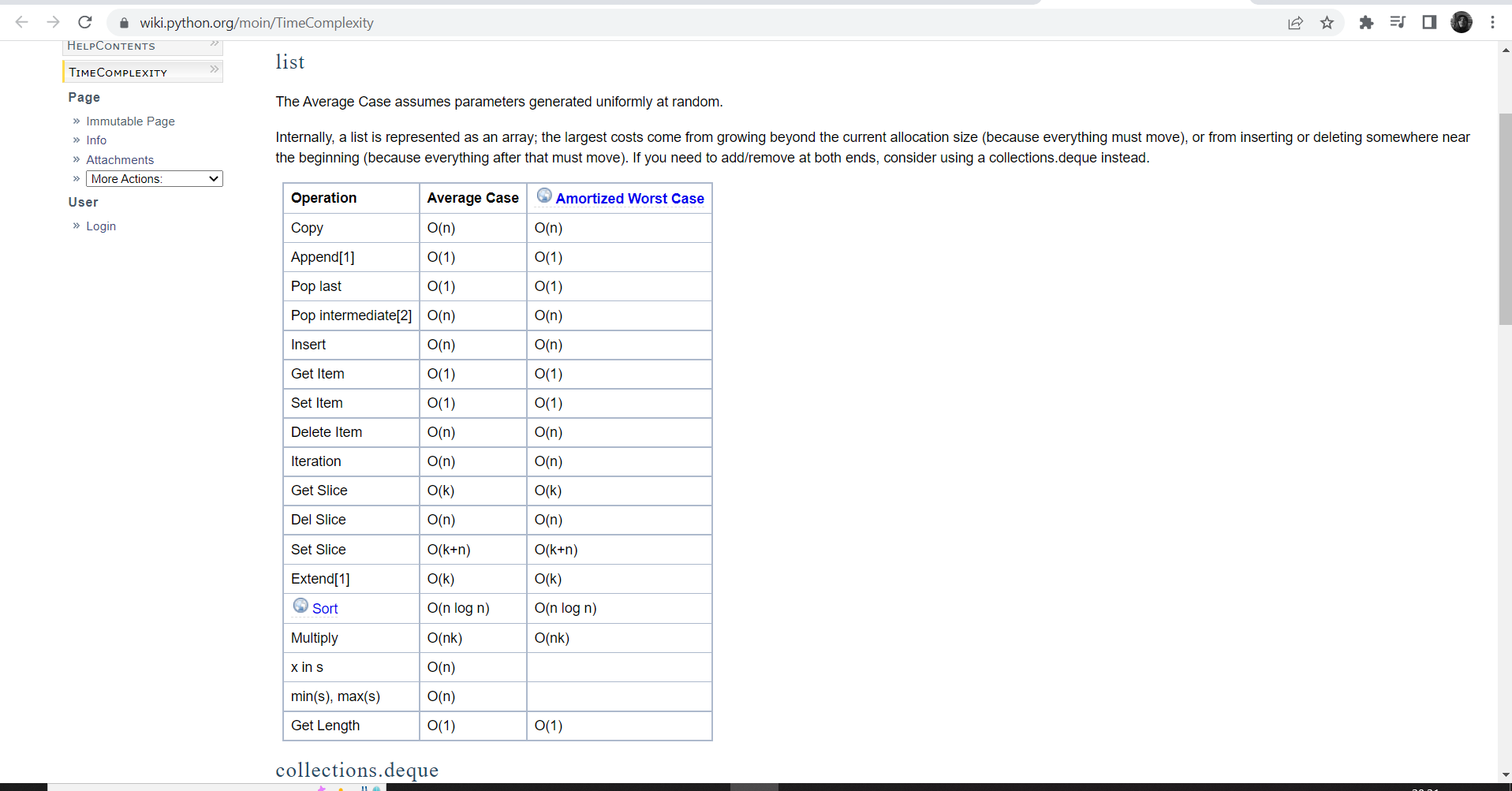
g. *T*(*n, k*) = 8*n* + *k* log *n* + 800 = O(n)

h. *T*(*n, k*) = 100*kn* + *n =* O(kn)

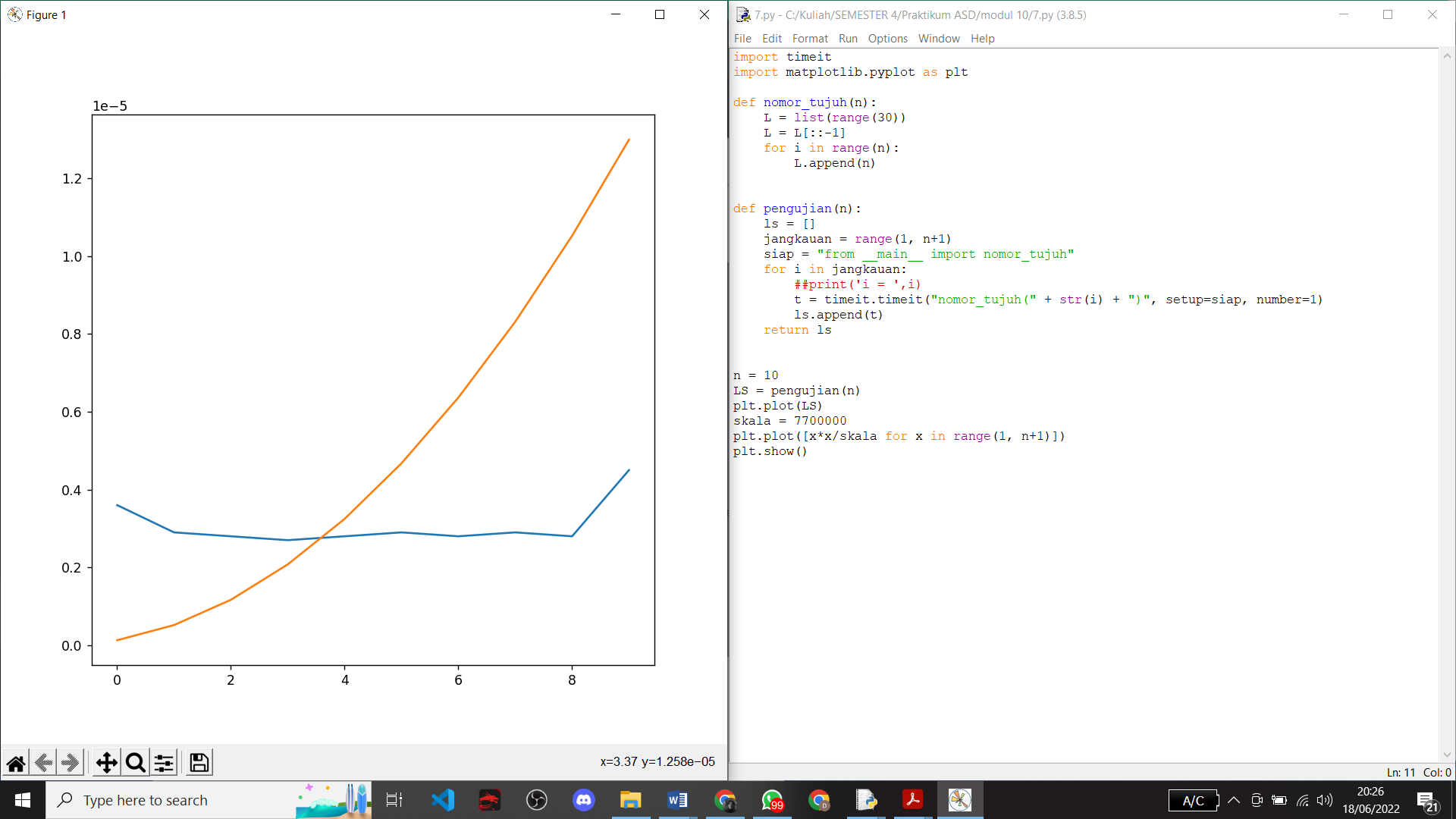
6. (Literatur Revew) carilah di internet, kompleksitas metode-metode pada object list di Python. Hint

- Google python list methods complexity. Lihat juga bagian ‘Images’-nya

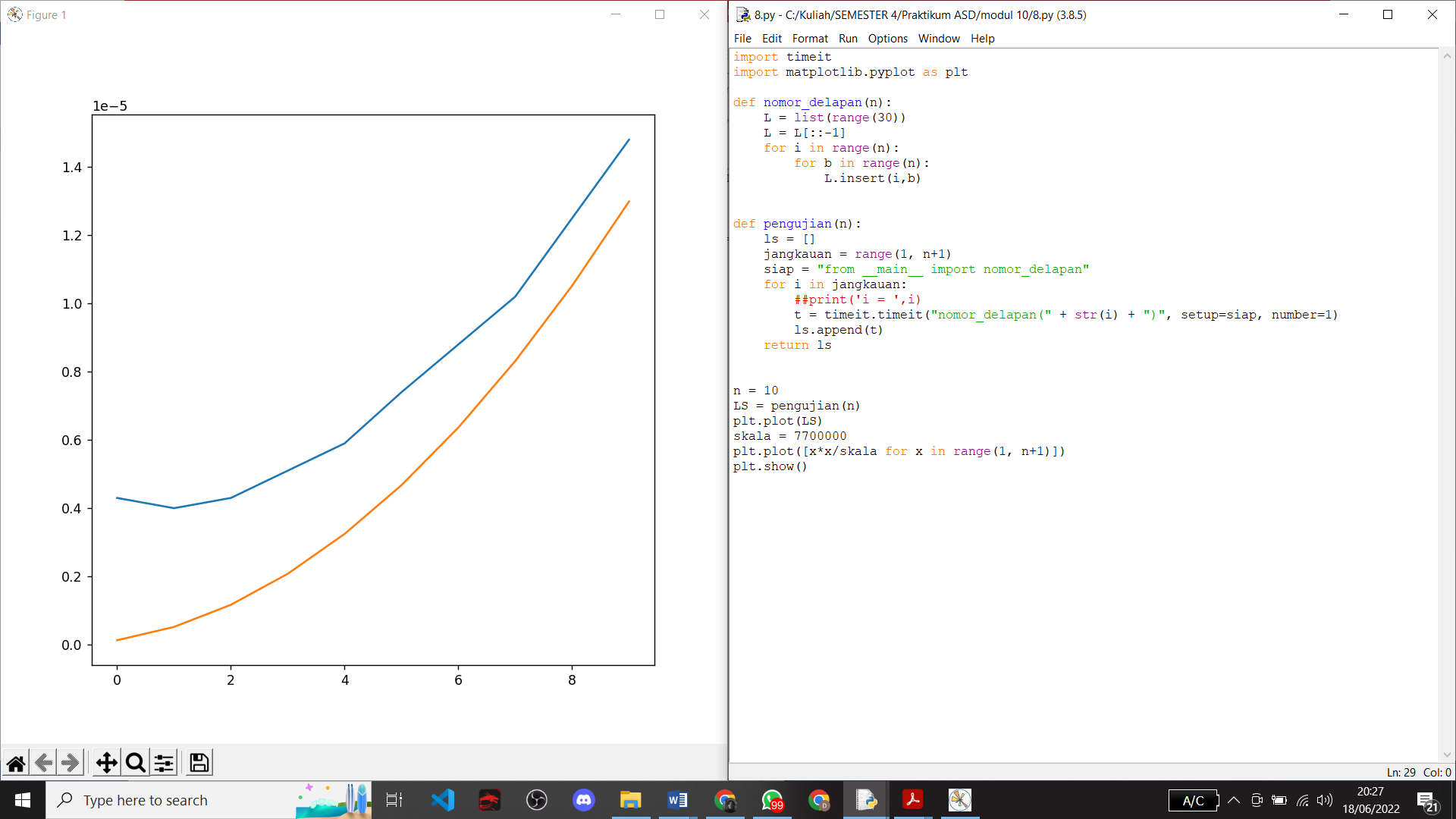
- Kunjungi https://wiki.python.org/moin/TimeComplexity



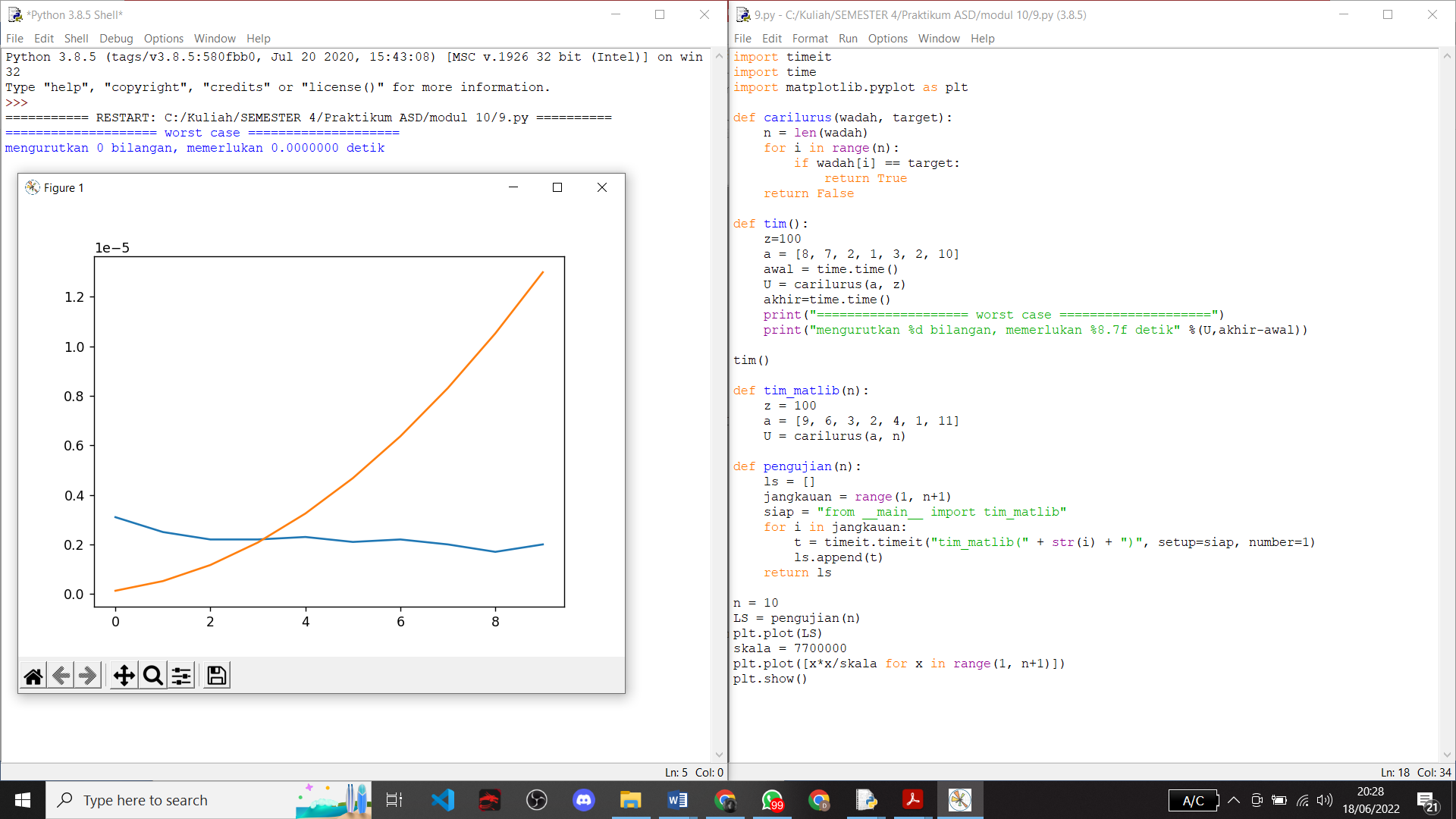
7.



8.



9.



10.



11.

* Big O dilambangkan dengan notasi O(...) merupakan keadaan terburuk (worst case). Kinerja seubuah algoritma biasanya diukur menggunakan patokan keadaan Big-O ini. Merupakan notasi asymptotic untuk batas fungsi dari atas dan bawah dengan Berperilaku mirip dengan ≤ operator untuk tingkat pertumbuhan.
* Big Theta dilmbangkan dengan notasi Θ(...) merupakan notasi asymptotic untuk batas atas dan bawah dengan keadaan terbaik (best case). Menyatakan persamaan pada pertumbuhan f (n) hingga faktor konstan (lebih lanjut tentang ini nanti). Berperilaku mirip dengan = operator untuk tingkat pertumbuhan.
* Big Omega dilambangkan dengan notasi Ω(...) merupakan notasi asymptotic untuk batas bawah dengan keadaan rata-rata(average case) yang berperilaku mirip dengan ≥operator untuk tingkat pertumbuhan.

12.

Amortized analysis adalah metode untuk menganalisis kompleksitas algoritma yang diberikan, atau berapa banyak resource nya terutama waktu atau memori yang diperlukan untuk mengeksekusi. Dapat ditunjukkan dengan waktu rata-rata yang diperlukan unyuk melakukan satu urutan operasi pada struktur data terhadap keseluruhan operasi yang dilakukan