



Zad 5 Przemysław Wieczorek; Dominik Lisiecki

- W wyznaczonym zadaniu naszym celem jest napisanie skryptu który będzie prezentacją algorytmu Smitha-Watermana. Program ma wyszukiwać wszystkie pod-sekwencję składające się z ponad K aminokwasów
- Wyszliśmy z założenia że skrypt będzie składał się z 4 sekcji :
 - - oceny macierzy
 - - funkcji obliczającej najlepsze dopasowanie algorytmu do zadanych wartości na podstawie śledzenia macierzy „punktów”
 - - funkcji do ukazania wyników

- NumPy
- Numba

Użyte biblioteki

Zbudowaliśmy 2 funkcje, aby przedstawić na 2 różne sposoby przedstawienia wyników. Możemy pokazać krok po kroku jak również tylko same przyrównania i wynik punktowy.

Sposoby wyświetlania danych wynikowych

- ▶ Widoczne jest, która sekwencja jest liczona
- ▶ Ilość przejść jest liczona
- ▶ Na końcu ukazuje się finalne dopasowanie po przeliczeniu i dopasowaniu

[illegible]

2 – pokazanie dopasowań

- Widoczne są wszystkie dopasowania
- Punktacja widoczna na samej górze
- Ukazana jest liczba wszystkich uzyskanych przyrównań

Wszystkie przyrównani: 72
Wynik punktacji: 20

-CTGAACTTAATTTCCCCAA-G-A-TGA
AATGAG-TTAATGTAG--AAGGCACT-A

C-TGAACTTAATTTCCCCAA-G-A-TGA
AATGAG-TTAATGTAG--AAGGCACT-A

-CTGAACTTAATTTCCCCAA-G-A-TGA
AATGA-GTTAATGTAG--AAGGCACT-A

C-TGAACTTAATTTCCCCAA-G-A-TGA
AATGA-GTTAATGTAG--AAGGCACT-A

-CTGAACTTAATTTCCCCAA-G-A-TGA
AATG-AGTTAATGTAG--AAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATGAG-TTAATGT-A-GAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATGAG-TTAATGT-A-GAAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATGA-GTTAATGT-A-GAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATGA-GTTAATGT-A-GAAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATG-AGTTAATGT-A-GAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATG-AGTTAATGT-A-GAAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATGAG-TTAATGT--AGAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATGAG-TTAATGT--AGAAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATGA-GTTAATGT--AGAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATGA-GTTAATGT--AGAAGGCACT-A

-CTGAACTTAATTTCCCCAAG--A-TGA
AATG-AGTTAATGT--AGAAGGCACT-A

C-TGAACTTAATTTCCCCAAG--A-TGA
AATG-AGTTAATGT--AGAAGGCACT-A

>>>



Znane problemy

- Działanie programu opartego na podejściu rekurencyjnym powoduje duże zużycie pamięci RAM
- Powolne przetwarzanie „dużych” sekwencji
- Niedostosowanie programu do tzw. „akceleracji sprzętowej”
- Ogólna powolność języka Python

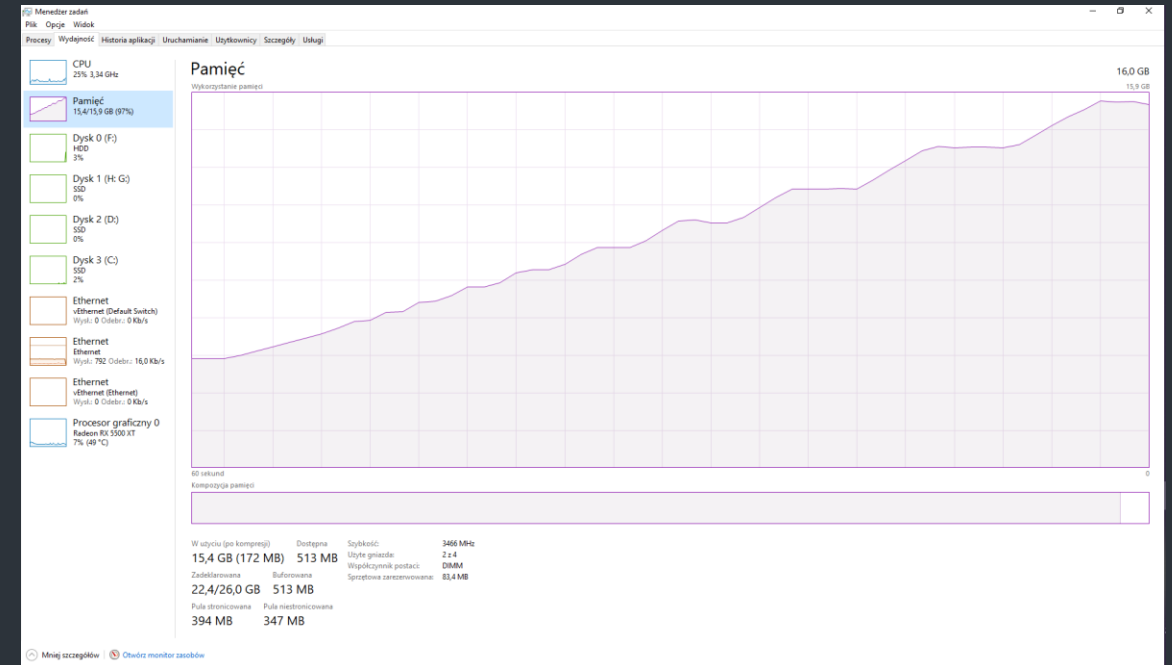
Przyczyna/Skutek

Zużycie pamięci RAM/MemoryError

Rekurencyjny sposób rozwiązania zadania, przy użyciu sekwencji o długości około 16600n (sekwencja genu mitochondrialnego *Vulpes zerda* i *Vulpes lagopus*) powoduje wzrost zużycia pamięci RAM osiągający limit fizyczny urządzenia.

Skutkiem tego jest, że w pewnym momencie dostajemy błąd MemoryError i program kończy działanie

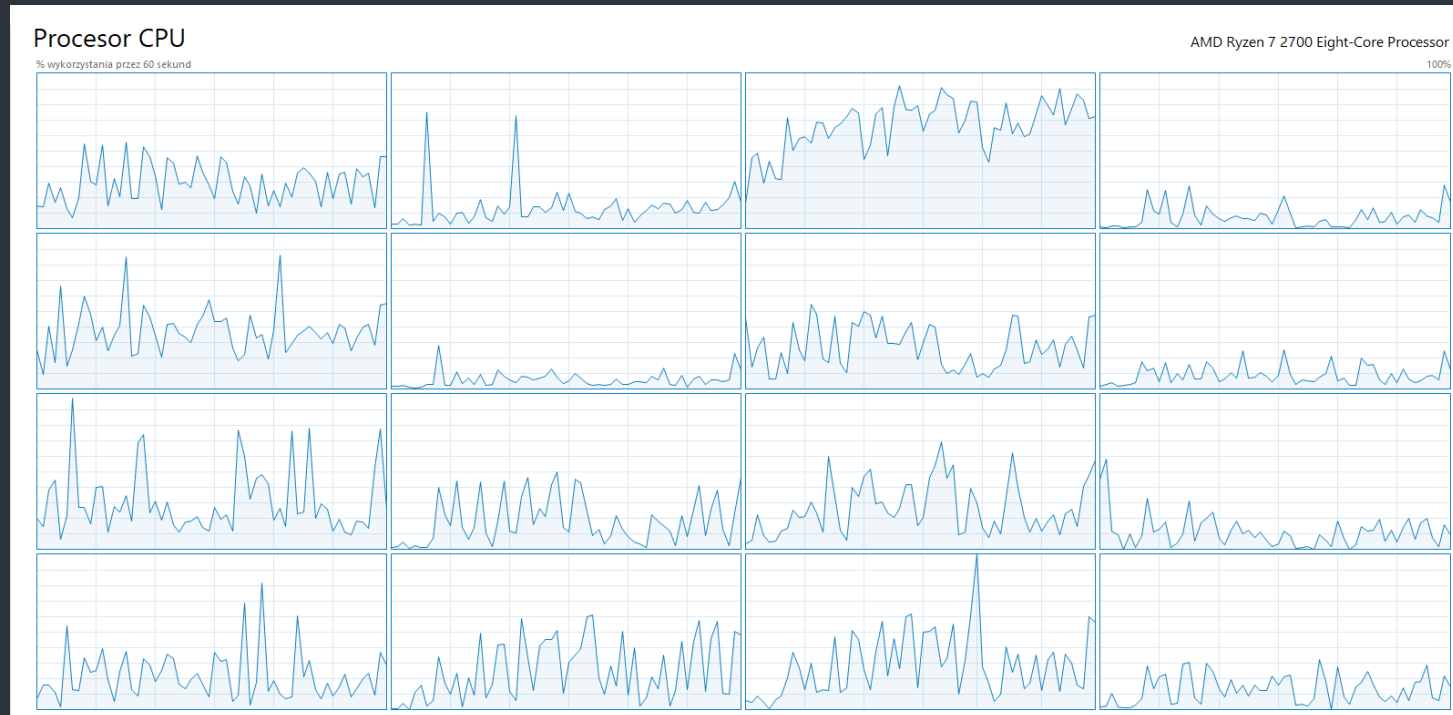
```
MemoryError  
>>>
```



Powolność programu/niewykorzystanie możliwości sprzętu

Samo zużycie ram'u nie jest jednym problemem. Dochodzi również to iż cały kod jest uruchamiany głównie na jednym rdzeniu logicznym i w niewielkim stopniu na 15 pozostałych.

Pod uwagę trzeba wziąć, że w trakcie działania były również włączone inne programy co przekłada się podane wykresy



Powolność programu/niewykorzystanie możliwości sprzętu

Niewiele, ale jednak pomocna okazała się biblioteka Numba. Pozwala ona przyspieszenie działania kodu poprzez zoptymalizowanie kodu maszynowego.

Jednocześnie, co można zaobserwować na wykresie pomogła częściowo rozłożyć działanie kodu na większość z 16 wątków.



Pomysły na dalsze rozwiązania?

- ▶ Przeniesienie kodu i przystosowanie pod bibliotekę PyTorch lub...
- ▶ Przeniesienie i przystosowanie do użycia biblioteki TensorFlow

Powyższe biblioteki pozwalają na uruchomienie danego zadania na wybranym urządzeniu, pozwala to na wykorzystanie GPU jako głównej jednostki obliczeniowej. Jednak aby jednostka była zdolna do tego wymaga GPU z wybranych serii od nVidii. Co ogranicza działanie z dwóch jednostek do jednej. Jeszcze jednym sposobem jest:

- ▶ Platforma AMD ROCm dla jednej z jednostek

Jednak są to już narzędzia głównie dla platformy HPC z którą styczności nie mamy jako studenci(nie na naszym uniwersytecie)

