

Lab. 4 – Wektory

Zadania z zajęć

1. Napisz program, który wczytuje w pętli kolejne liczby i zapisuje w wektorze. Długość wektora podana jest przez użytkownika. Program wyświetla na koniec cały wektor

Dane wejściowe: N , $w[0]$, $w[1]$, $w[2]$, ... , $w[N-1]$

Wyświetlany wynik: w

Zmodyfikuj program tak, żeby wczytywał cały wektor jednym poleceniem (bez użycia pętli) i wyświetlał jego długość.

2. Napisz program, który oblicza średnią elementów wektora. Wektor podawany przez użytkownika.

Dane wejściowe: w

Wyświetlany wynik: średnia z wartości z w

3. Napisz program, który oblicza wariancję wektora. *(na zajęciach, lub do domu)

Dane wejściowe: w

Wyświetlany wynik: wariancja z wartości z w

4. Napisz program, który znajduje wartość maksymalną w wektorze.

Dane wejściowe: w

Wyświetlany wynik: maksimum z w

5. Napisz program, który oblicza sumę dwóch wektorów. Program powinien najpierw sprawdzić, czy sumowanie wektorów jest możliwe

Dane wejściowe: wektory a , b

Wyświetlany wynik: wektor równy $a + b$

6. Napisz program, który oblicza iloczyn skalarny dwóch wektorów. Program powinien najpierw sprawdzić, czy obliczenie iloczynu wektorów jest możliwe. *(na zajęciach, lub do domu)

Dane wejściowe: a , b

Wyświetlany wynik: wartość $a \cdot b$

Dlaczego korzystamy z tablic numpy:

- W odróżnieniu od list tablice muszą zawierać elementy jednego typu – efektywne wykorzystanie pamięci

- Z góry ustalony rozmiar tablic (listy mogą być wydłużane i skracane) – szybsze działanie, lepsze wykorzystanie pamięci
- Numpy zawiera duży zestaw funkcji matematycznych, które można stosować na tablicach
- Szybsze operacje na dużych tablicach niż na dużych listach, lepsze wykorzystanie pamięci

Podsumowując, listy Pythona lepiej nadają się do ogólnych zadań wymagających elastyczności, natomiast tablice NumPy sprawdzają się w obliczeniach numerycznych i zadaniach obejmujących duże zbiory danych ze względu na swoją wydajność i zoptymalizowane działanie

Poznane na zajęciach

- `wektor.shape()`, `wektor.size()`, `len(wektor)`,
- `np.sum(wektor)`, `np.mean(wektor)`, `np.var(wektor)`
- `np.array(json.loads(input('podaj cały wektor (w postaci listy)')))`
- `w[3]` ← sposób indeksowania – dostęp do komórki czwartej wektora w

Pytania podsumowujące

- Jakie wartości zostaną wyświetlone na ekranie po uruchomieniu poniższego programu?

```
1 import numpy as np
2
3 w = np.array([1, 9, 8, 0, 2, 9, 4])
4 a = np.zeros(w.shape[0])
5
6 for i in range(w.shape[0]):
7     if i%2 == 0:
8         a[i] = w[i]
9
10
11 print(a)
```

- Jaki błąd popełniono w poniższym programie?

```
1 N = int(input('podaj dlugosc wektora '))
2 w = zeros(N)
```

Zadania domowe

1. Napisz program, który znajduje wartość maksymalną w wektorze oraz pozycję (indeks) tej wartości w wektorze

Dane wejściowe: w

Wyświetlany wynik: maksimum z w i indeks maksimum

2. Napisz program, który oblicza średnią elementów wektora, a następnie wyznacza ile wartości w wektorze jest większych, a ile mniejszych od średniej.

Dane wejściowe: w

Wyświetlany wynik: średnia z w, liczba wartości z w mniejszych od średniej z w, liczba wartości z w większych od średniej z w

