PPPD - Lab. 06

Zaimplementuj samodzielnie proste algorytmy sortowania przez porównywanie omawiane na wykładzie:

- bąbelkowe,
- przez wstawianie,
- przez wybór,

służące do sortowania danej listy w miejscu.

Dla funkcji sortujących w miejscu zwracaj liczbę porownań i przestawień w postaci 2-elementowej tablicy.

1. Porównaj empirycznie liczbę porównań i przestawień w przypadku optymistycznym i pesymistycznym dla ciągów liczbowych różnych długości.
2. Ponadto wygeneruj 100 pseudolosowych n-elementowych ciągów i policz średnią liczbę wykonanych porównań oraz przestawień.

Uwaga:

- Rozważ różne n, np. ze zbioru $\{10, 20, 30, \ldots, 100\}$.
- Wyniki możesz przedstawić na wykresie, wywołując `plt.plot(lista_n, lista_wartosci)` (podobnie jak w innych zadaniach przygotowawczych). Dzięki temu zaobserwujesz, jak liczba porównań i przestawień rośnie w zależności od n – jest to ciekawy eksperyment zwłaszcza dla ciągów losowych.

Postaraj się, aby kod był dobrze zorganizowany – powtarzające się czynności zamknij w dobrze zdefiniowane, proste funkcje.

Co więcej, zaimplementuj powyższe algorytmy tak, by zwracały permutację sortującą.

Następnie zaimplementuj algorytm wyszukiwania połówkowego (binarnego) `binsearch(t, p, v)`, który znajduje indeks i taki, że $v = t[i]$ przy założeniu, że p jest permutacją sortującą elementy t, a kolejność elementów w t jest dowolna.

- Przetestuj jego działanie dla losowych `t` oraz `v`.