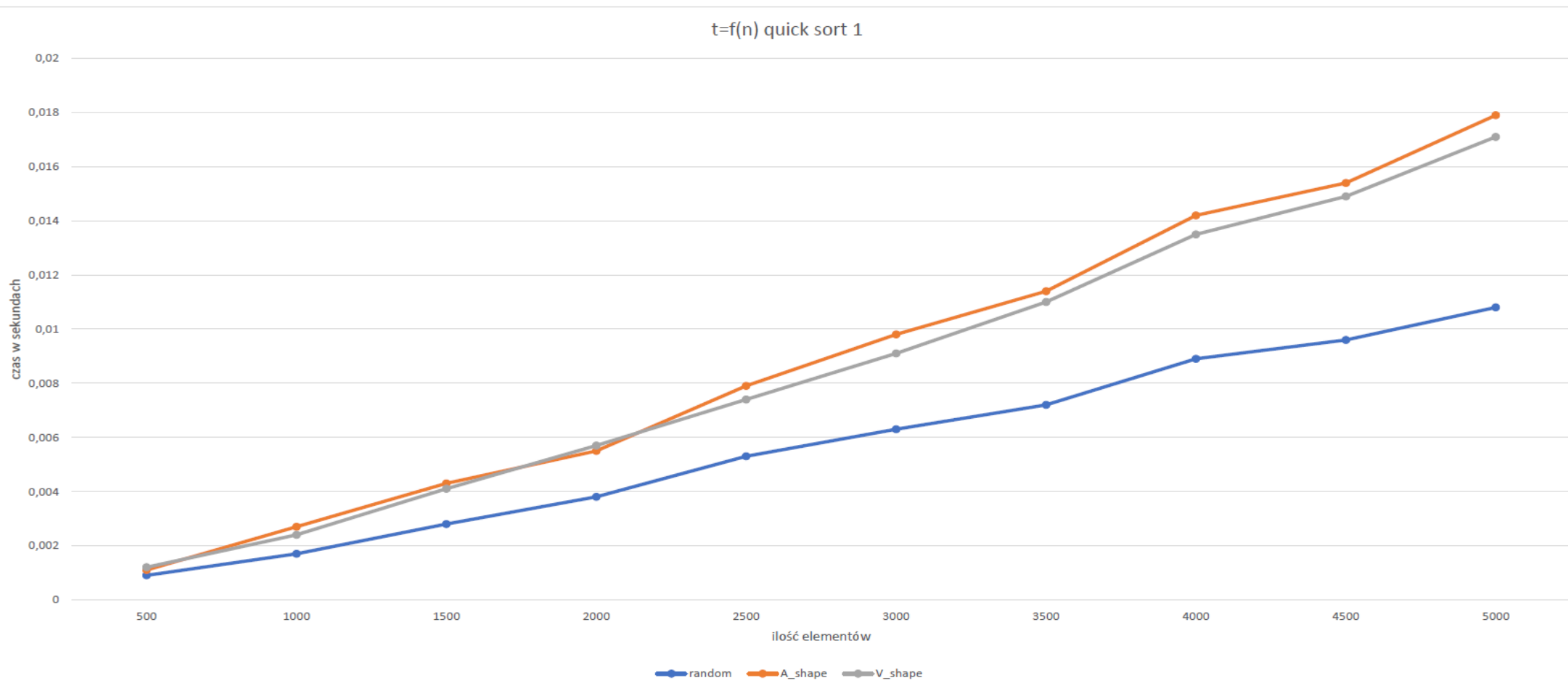
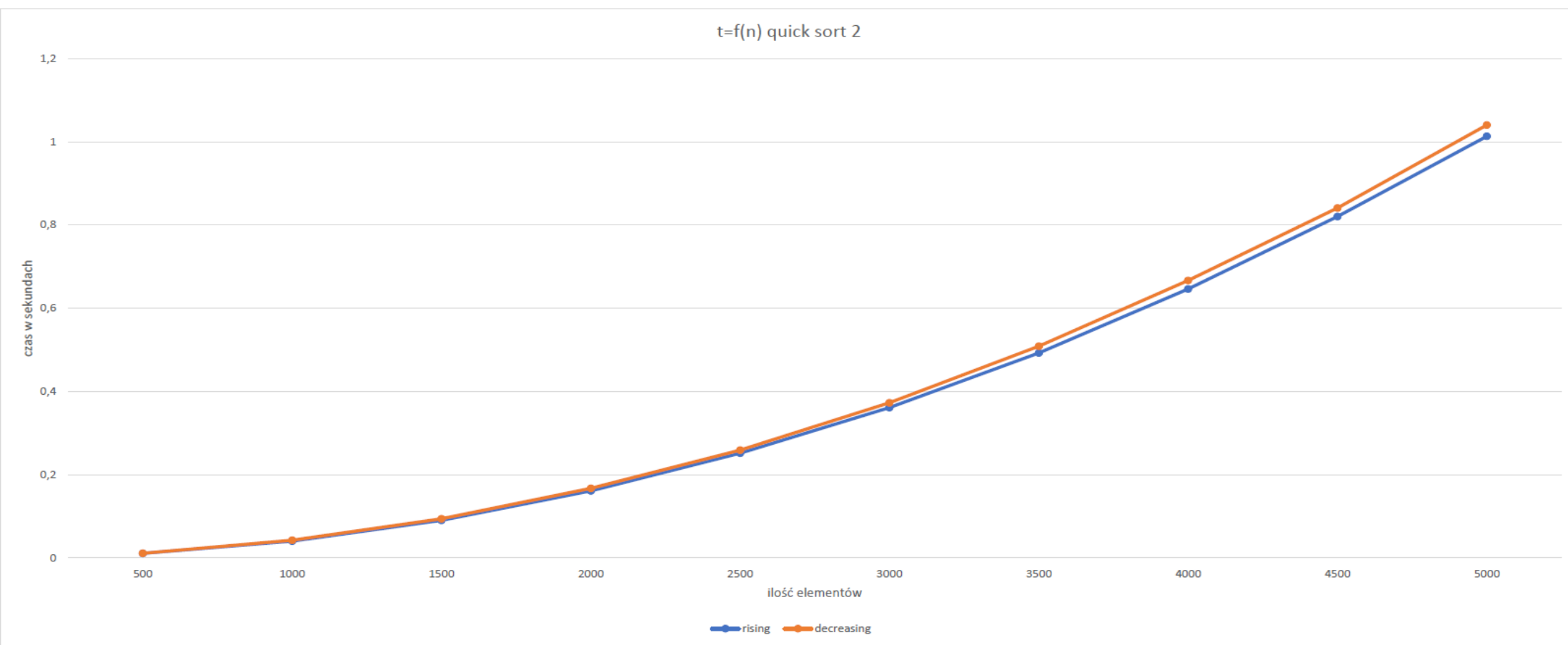


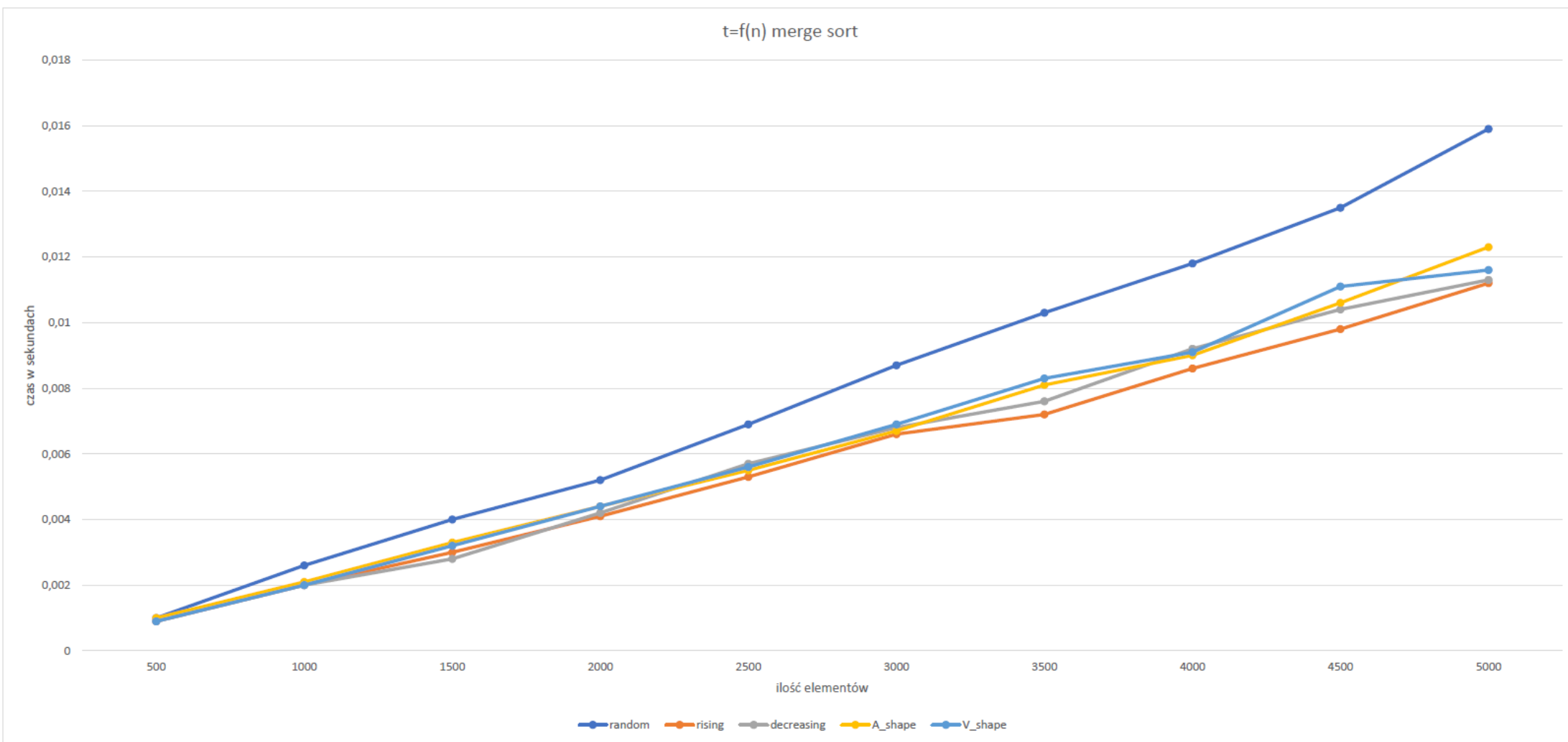
1) Poniższe wykresy prezentują zależność czasu wykonywania od ilości i rodzaju danych dla każdego z algorytmów sortowania.

1a) Wykres dla quick sort wymagał podzielenia go na dwa wykresy ze względu na dysproporcję czasu wykonywania dla danych rosnących i malejących w których algorytm ten wypadał bardzo niekorzystnie. Dla 5000 elementów różnica czasowa jest na poziomie dwóch rzędów wielkości.

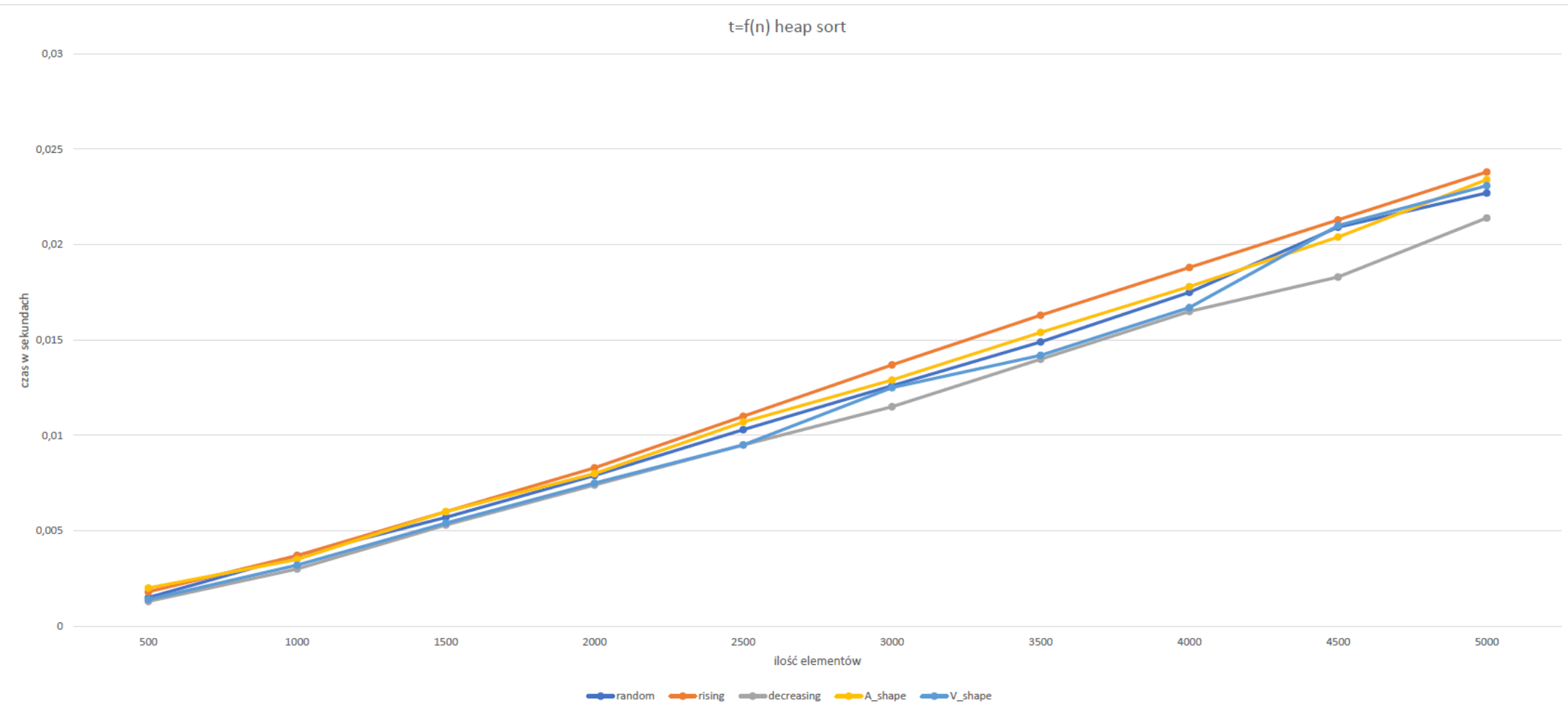




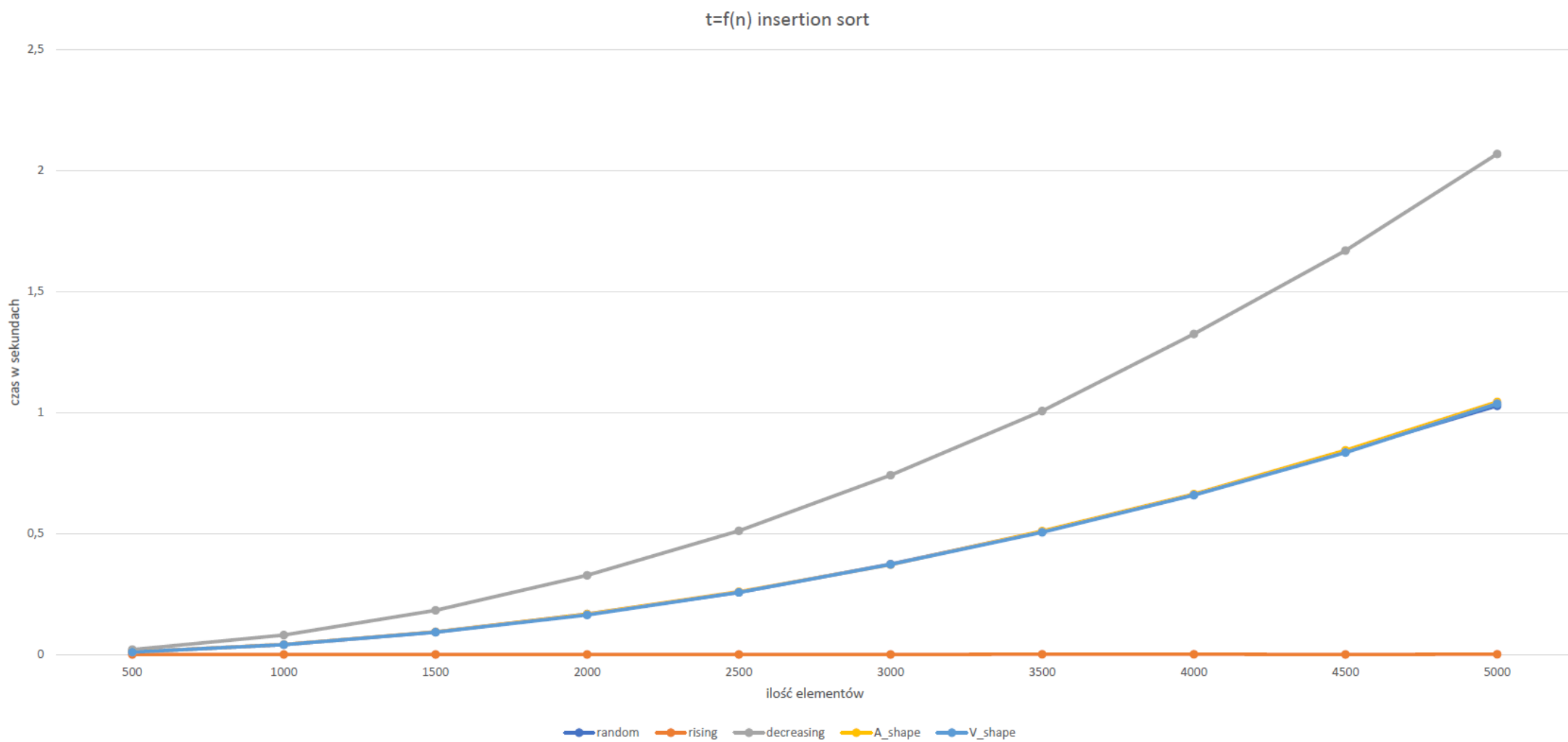
1b) Merge sort radzi sobie dobrze dla każdego typu danych, chociaż wyróżnić można dłuższy czas wykonywania dla danych losowych.



1c) Heap sort radzi sobie dobrze dla każdego typu danych.

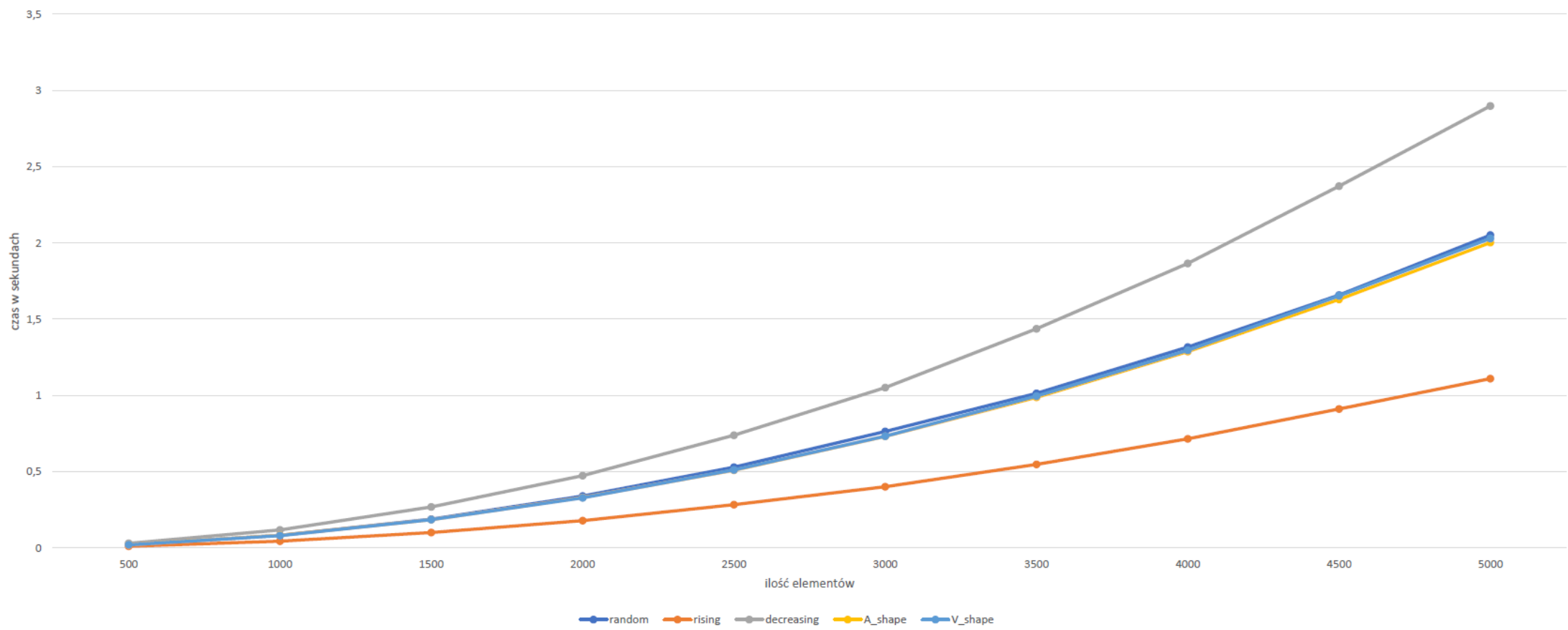


1d) Insertion sort jest algorytmem, który najlepiej ze wszystkich radzi sobie z danymi które już są posortowane. Wykres pokrywa się dla danych A kształtnych i V kształtnych. Najgorzej wypada dla danych malejących. Z wykresu wywnioskować można, że algorytm radzi sobie tym lepiej im więcej elementów jest już w porządkanej kolejności.

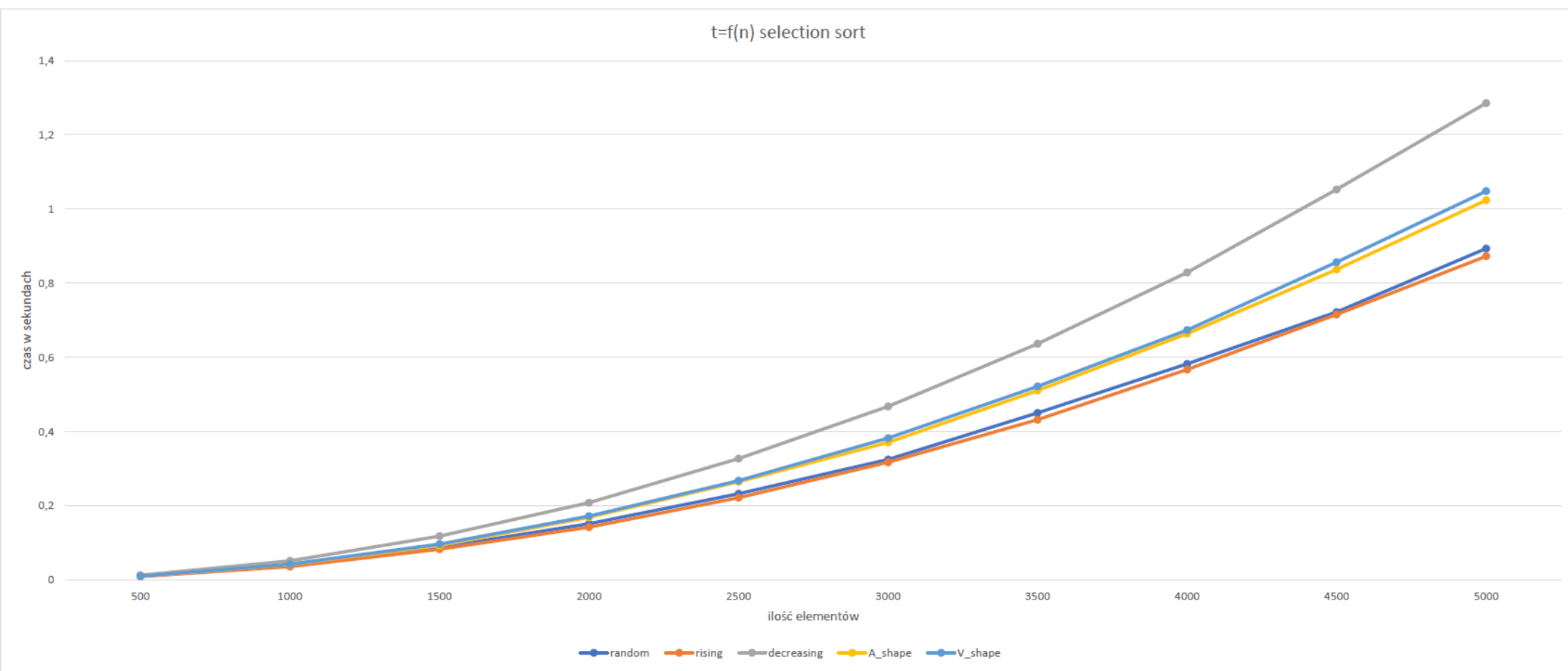


1e) Bubble Sort również lepiej radzi sobie z danymi, które ułożone w oczekiwanej kolejności.

t=f(n) bubble sort



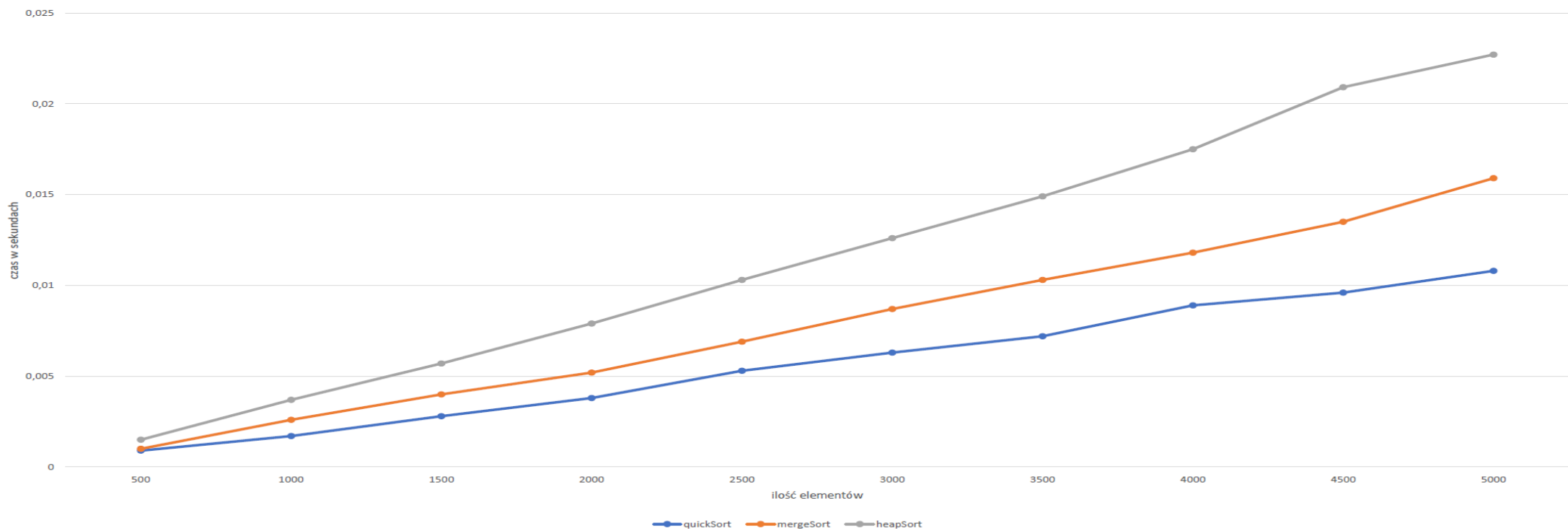
1f) Selection Sort.



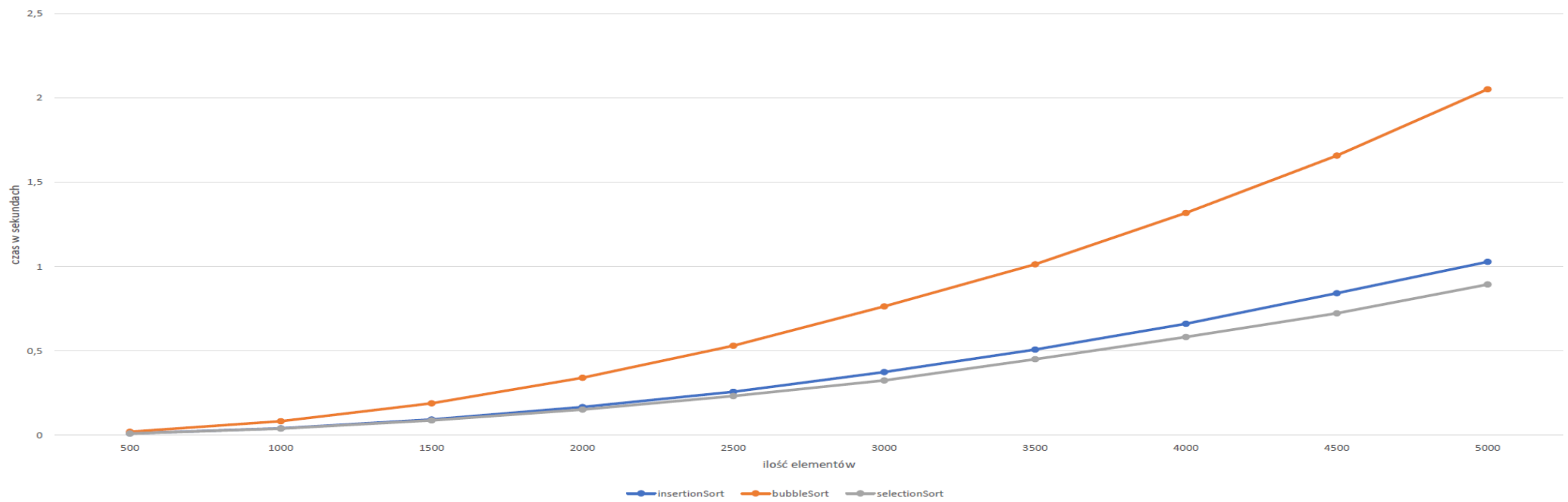
2) Poniższe wykresy porównują szybkość działania algorytmów dla danego typu danych

2a) Dane losowe. Ponownie dane muszą zostać przedstawione na dwóch wykresach (na jednym nie byłoby to dobrze widoczne) ze względu na niekorzystny czas wykonywania algorytmów naiwnych.

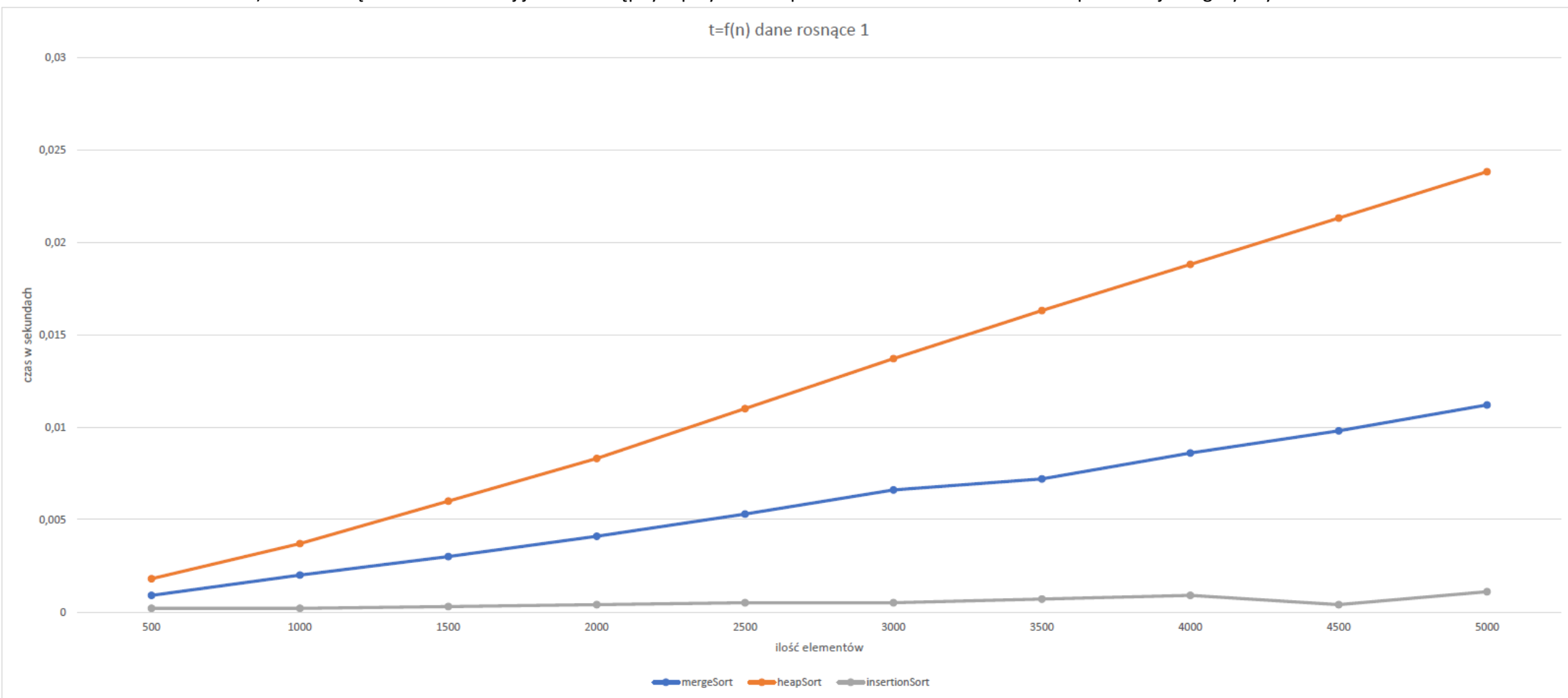
t=f(n) dane losowe 1

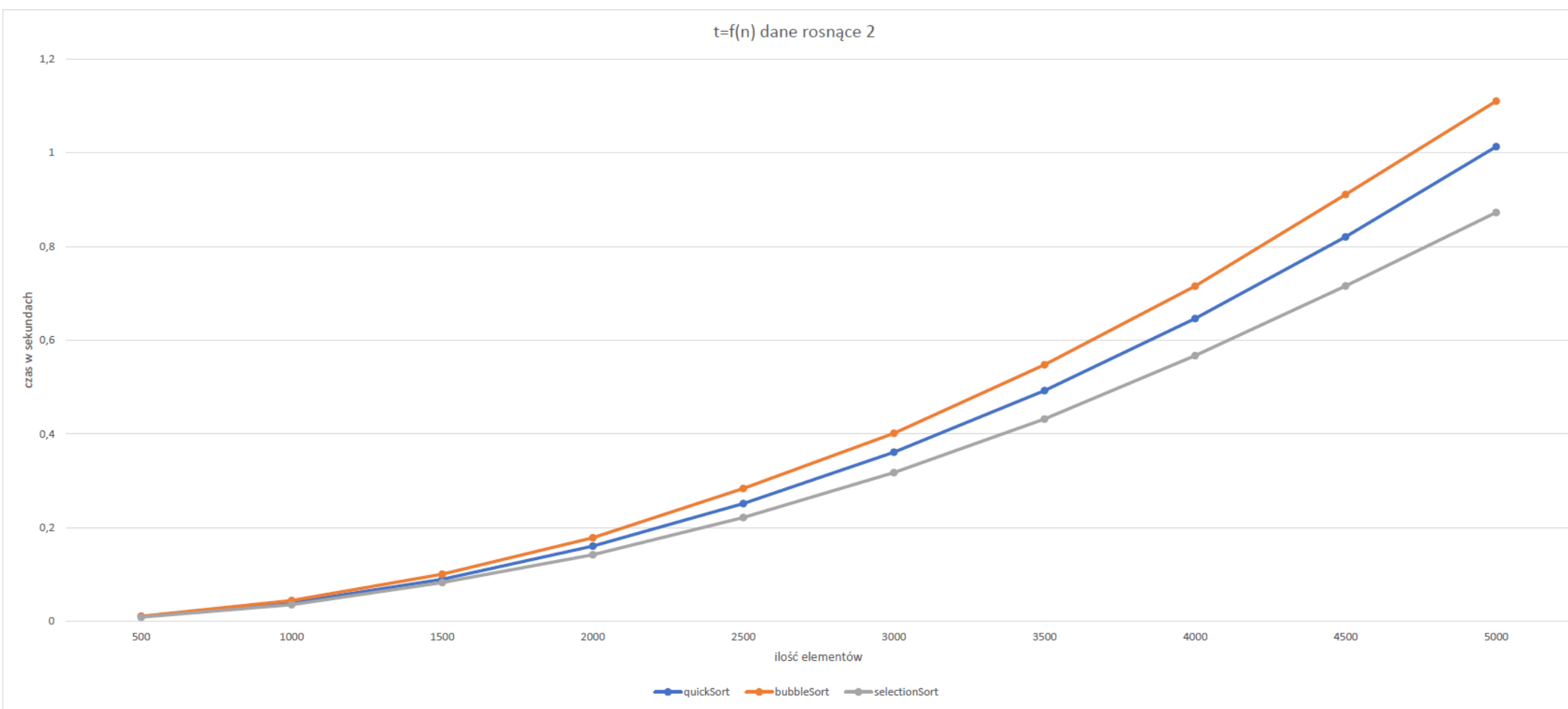


t=f(n) dane losowe 2



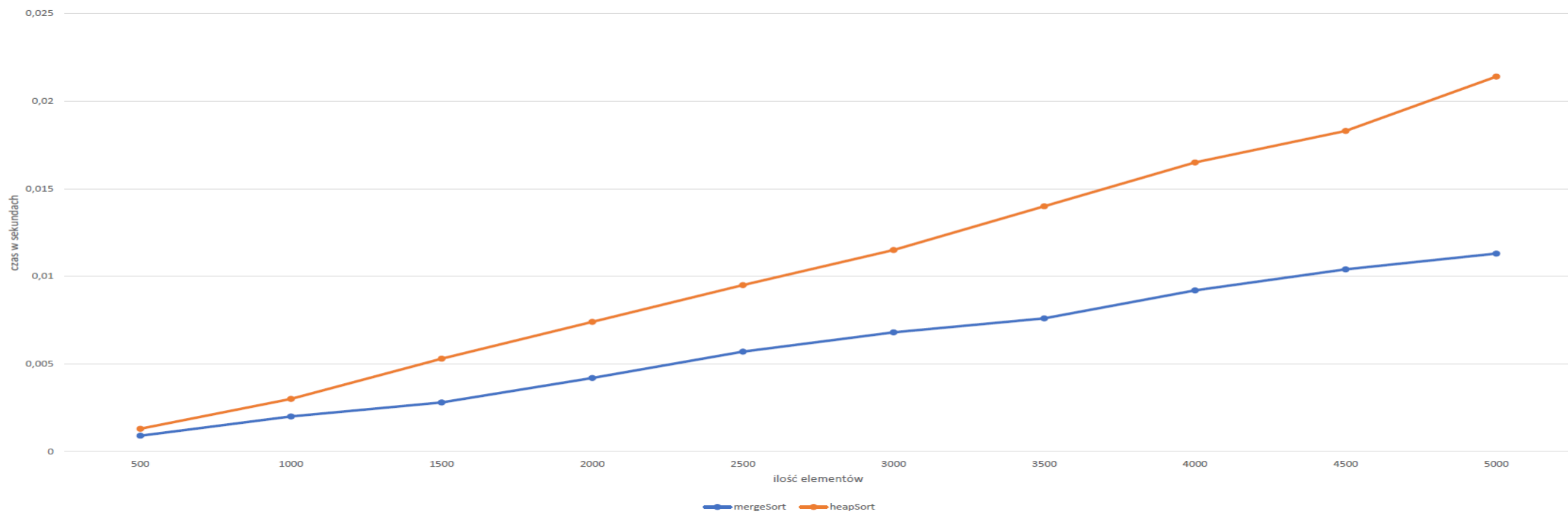
2b) Dane rosnące. Zarówno tutaj jak i w następnym przykładzie quick sort radzi sobie tak samo nieporadnie jak algorytmy naiwne



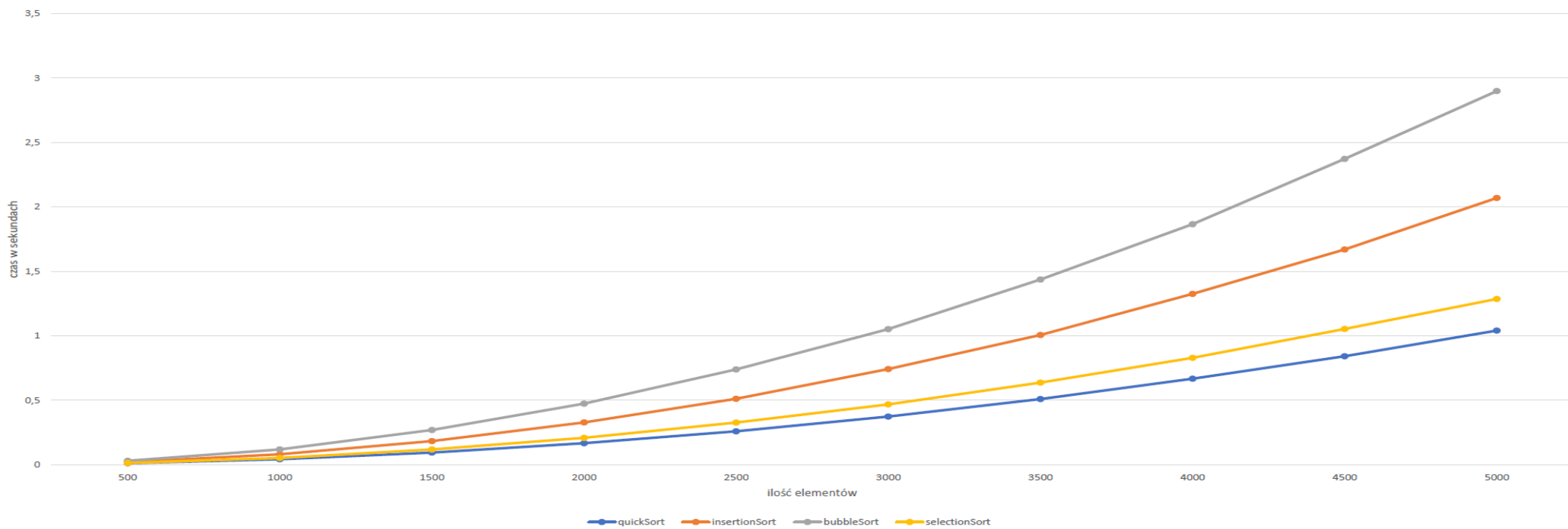


2c) Dane malejące.

t=(fn) dane malejące 1

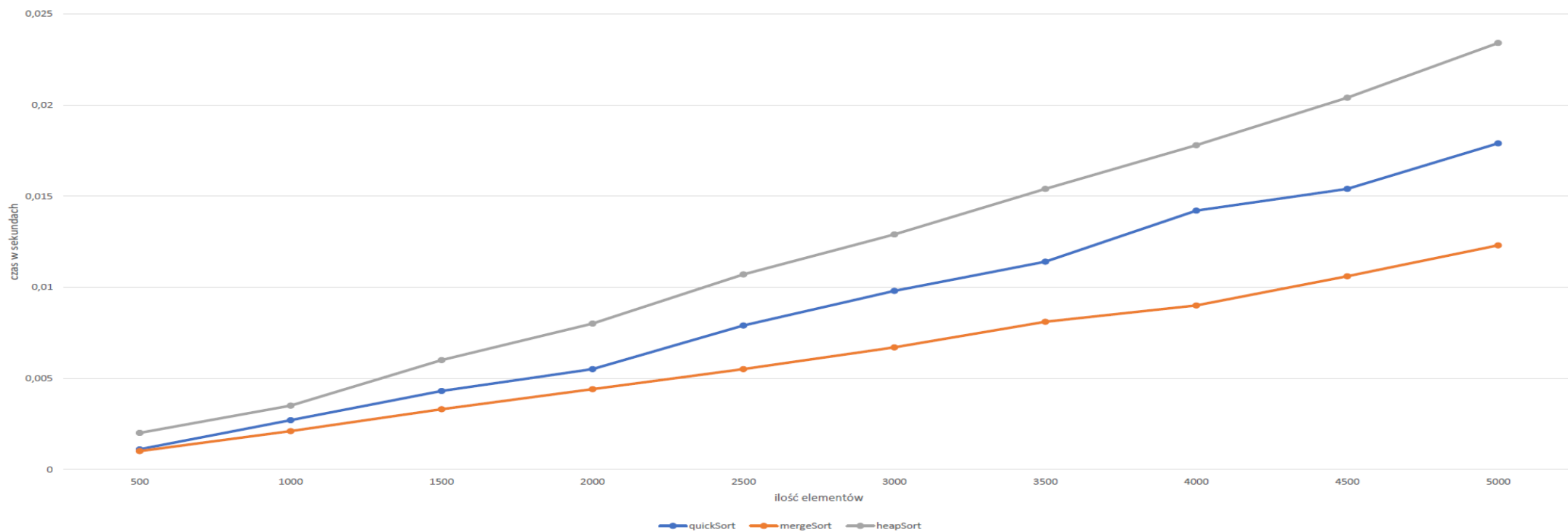


t=f(n) dane malejące 2

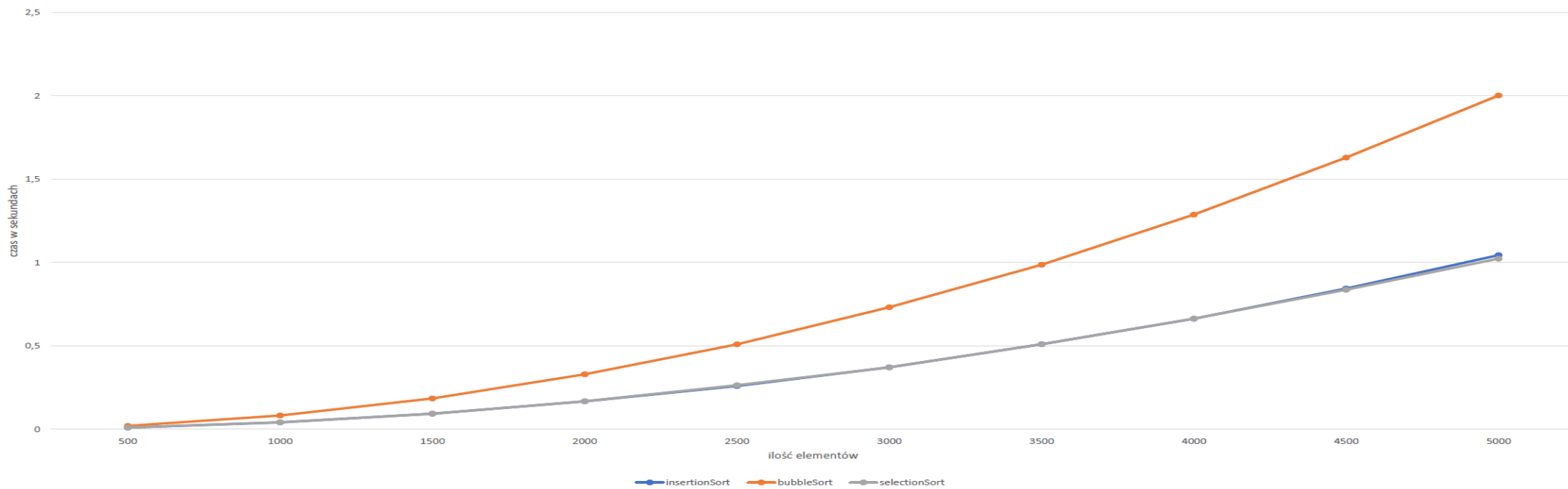


2d) Dane A kształtne.

$t=f(n)$ dane A kształtne 1

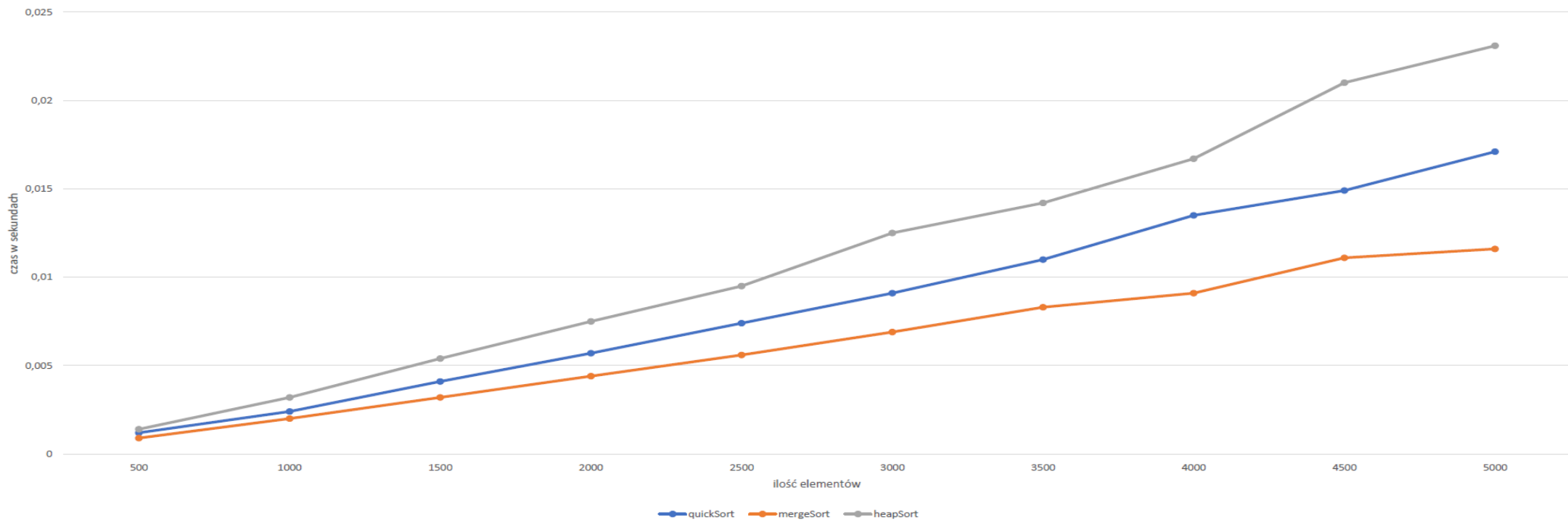


$t=f(n)$ dane A kształtne 2

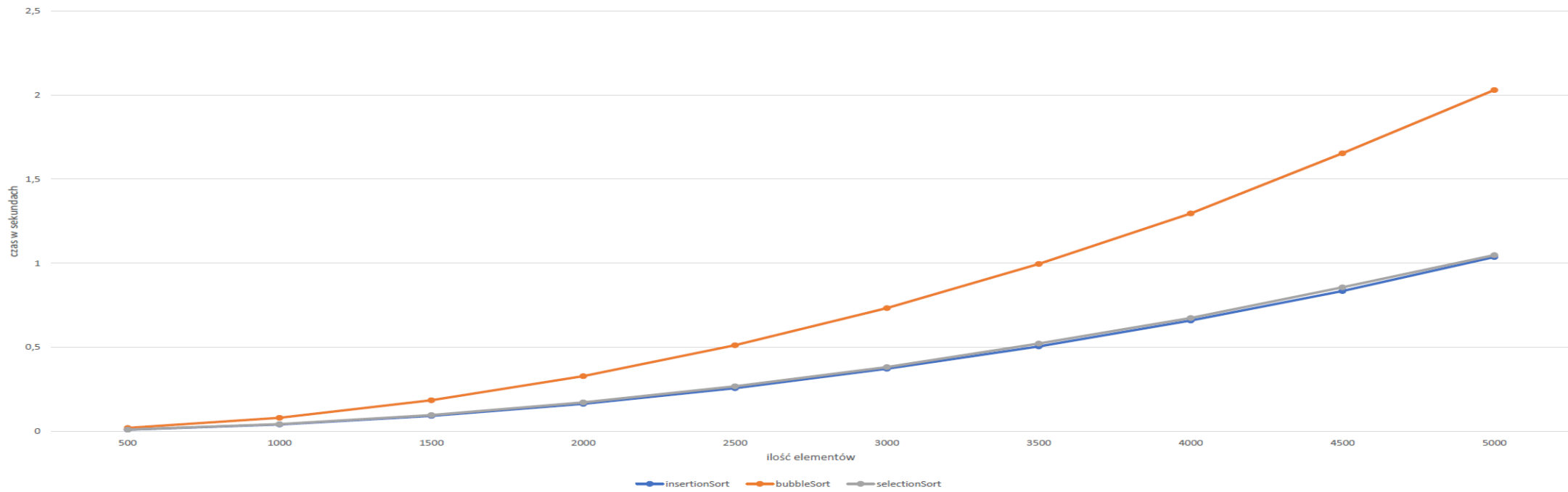


2e) Dane V kształtne.

$t=f(n)$ dane V kształtne 1

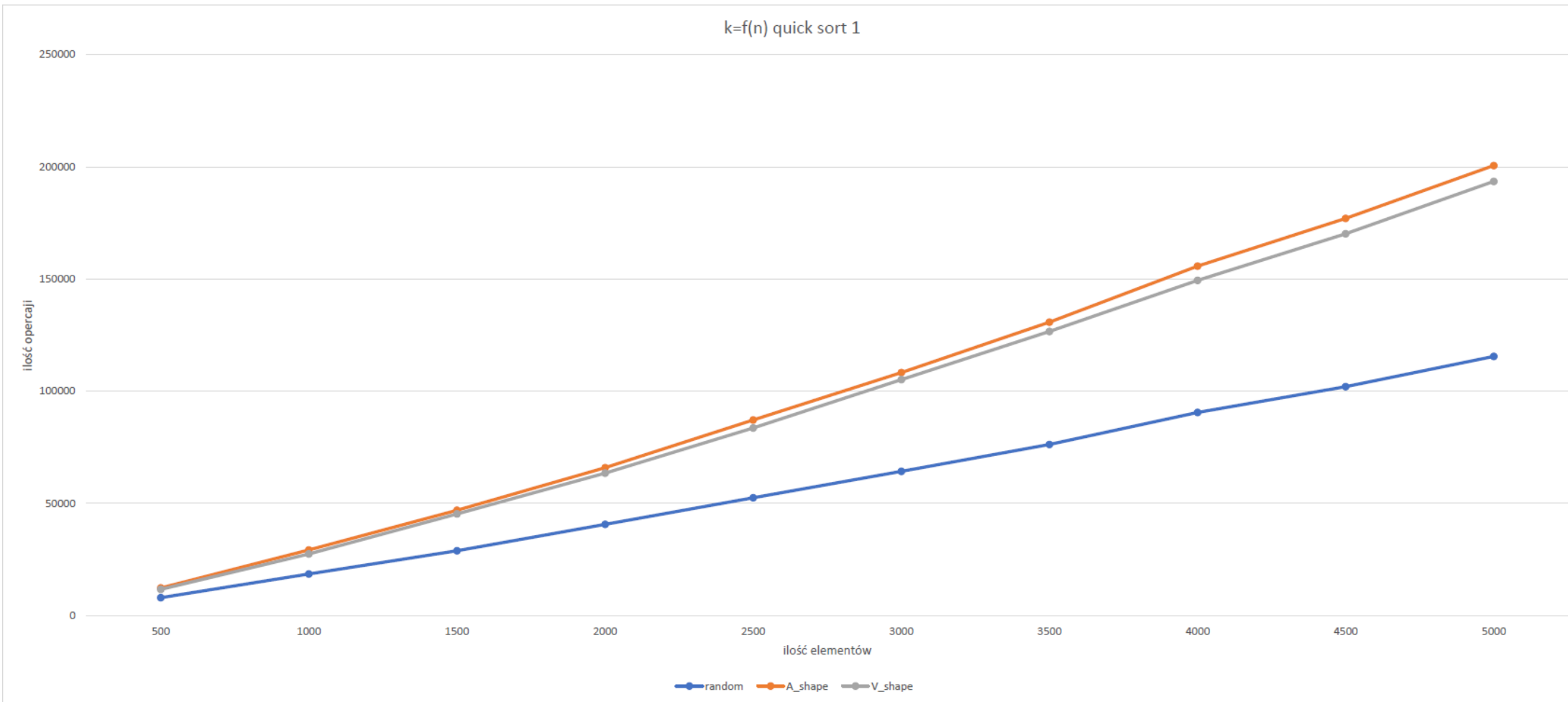


$t=f(n)$ dane V kształtne 2

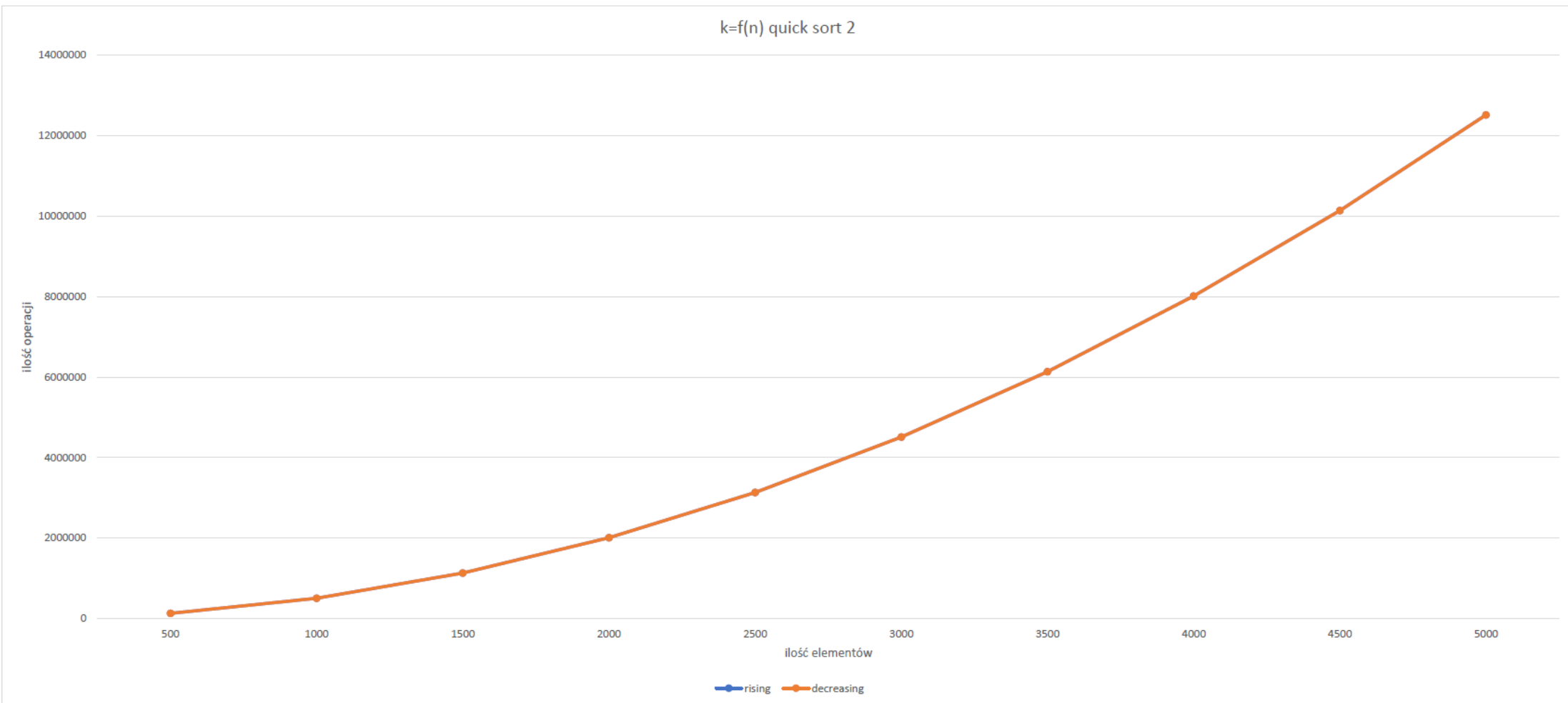


3) Poniższe wykresy przedstawiają jak poszczególne algorytmy radzą sobie w kontekście ilości operacji dla różnych typów i ilości danych wejściowych.

3a) Quick sort. Ponownie zauważyć można nieefektywność quick sortu dla danych rosnących i malejących.

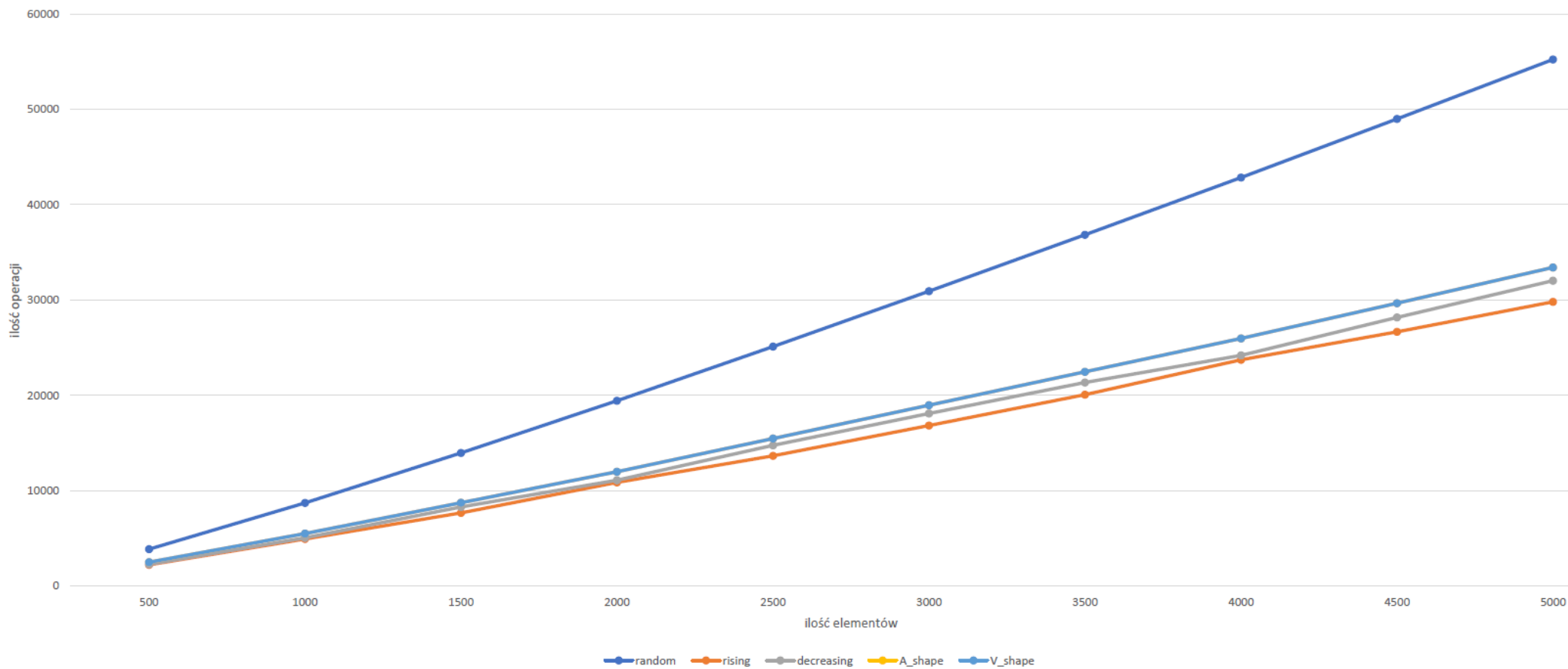


Wykres dla danych rosnących i malejących pokrywa się.

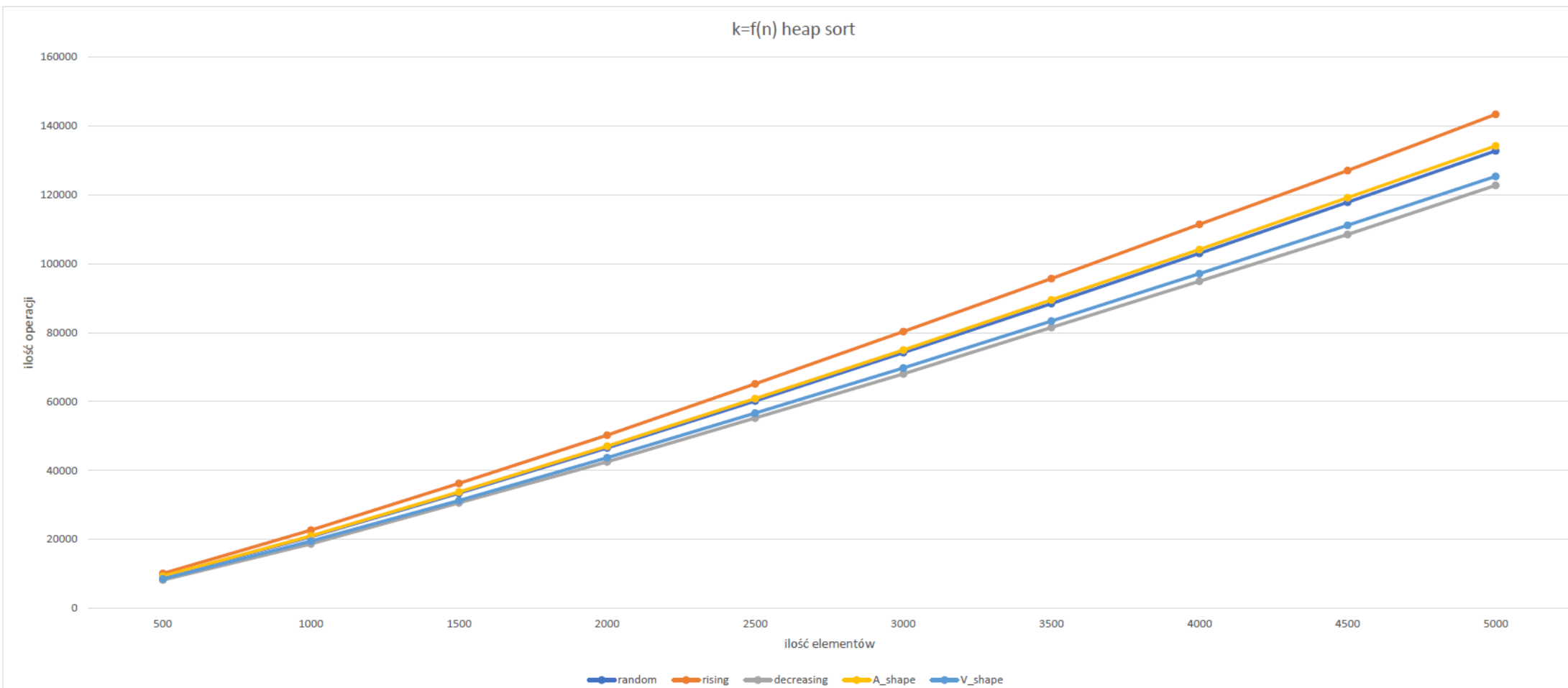


3b) Merge sort. Wykres dla danych A kształtnych i V kształtnych pokrywa się.

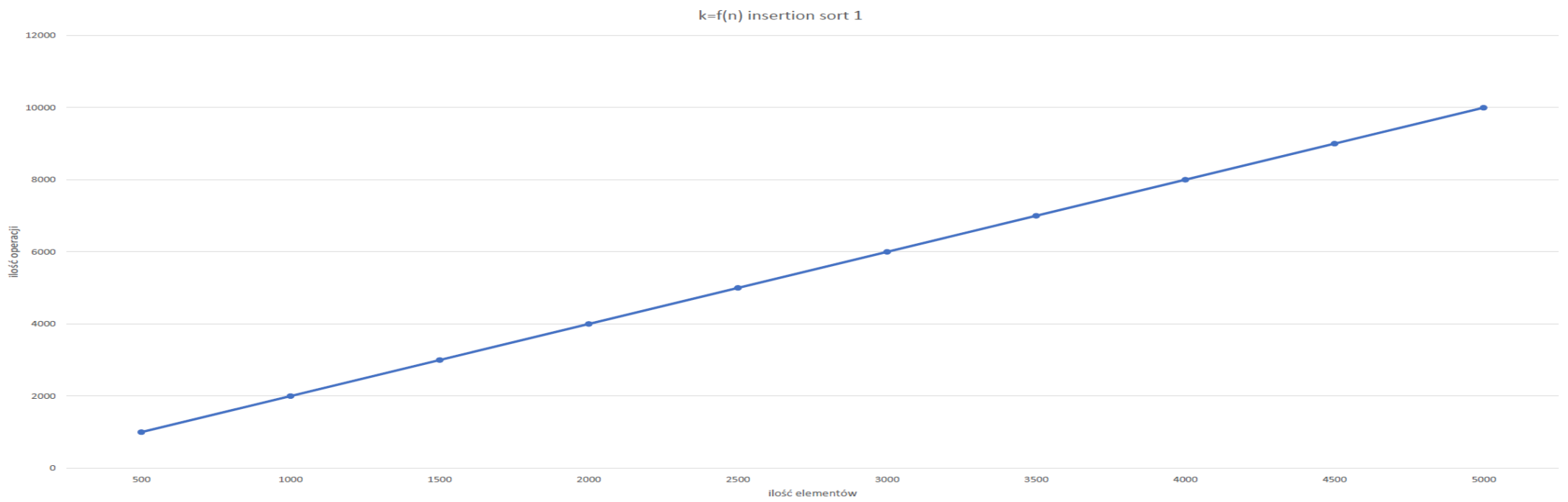
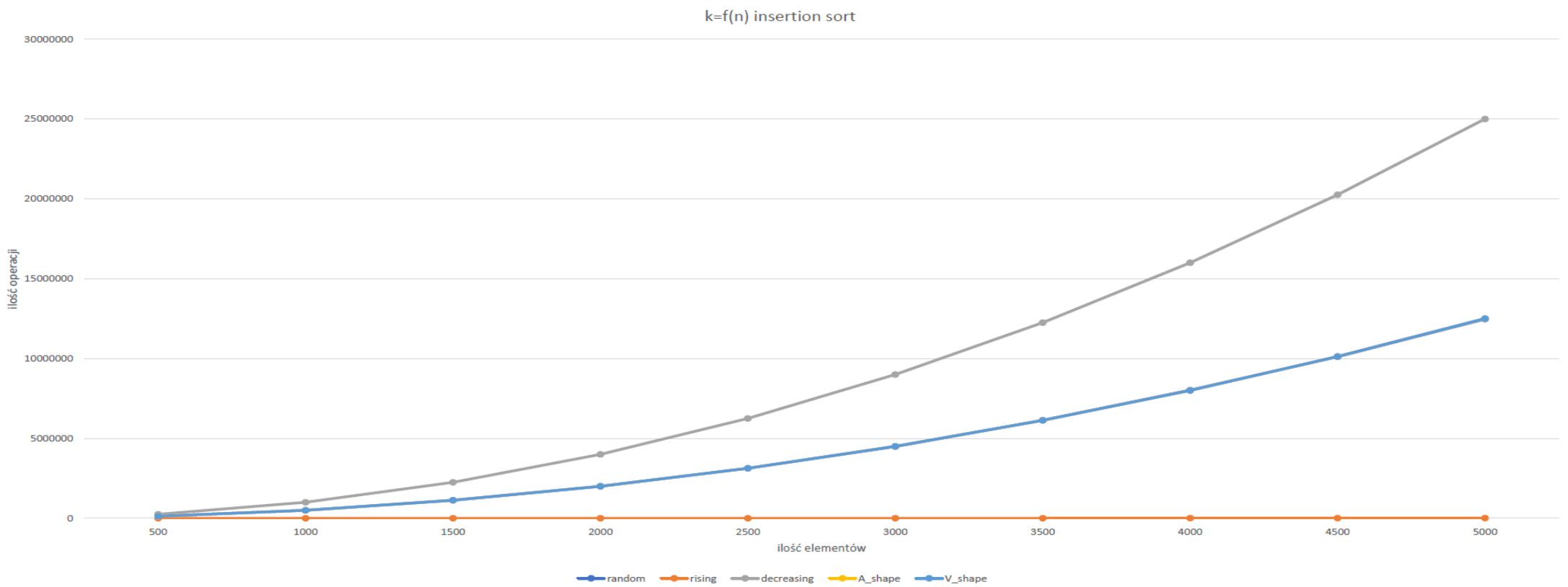
k=f(n) merge sort



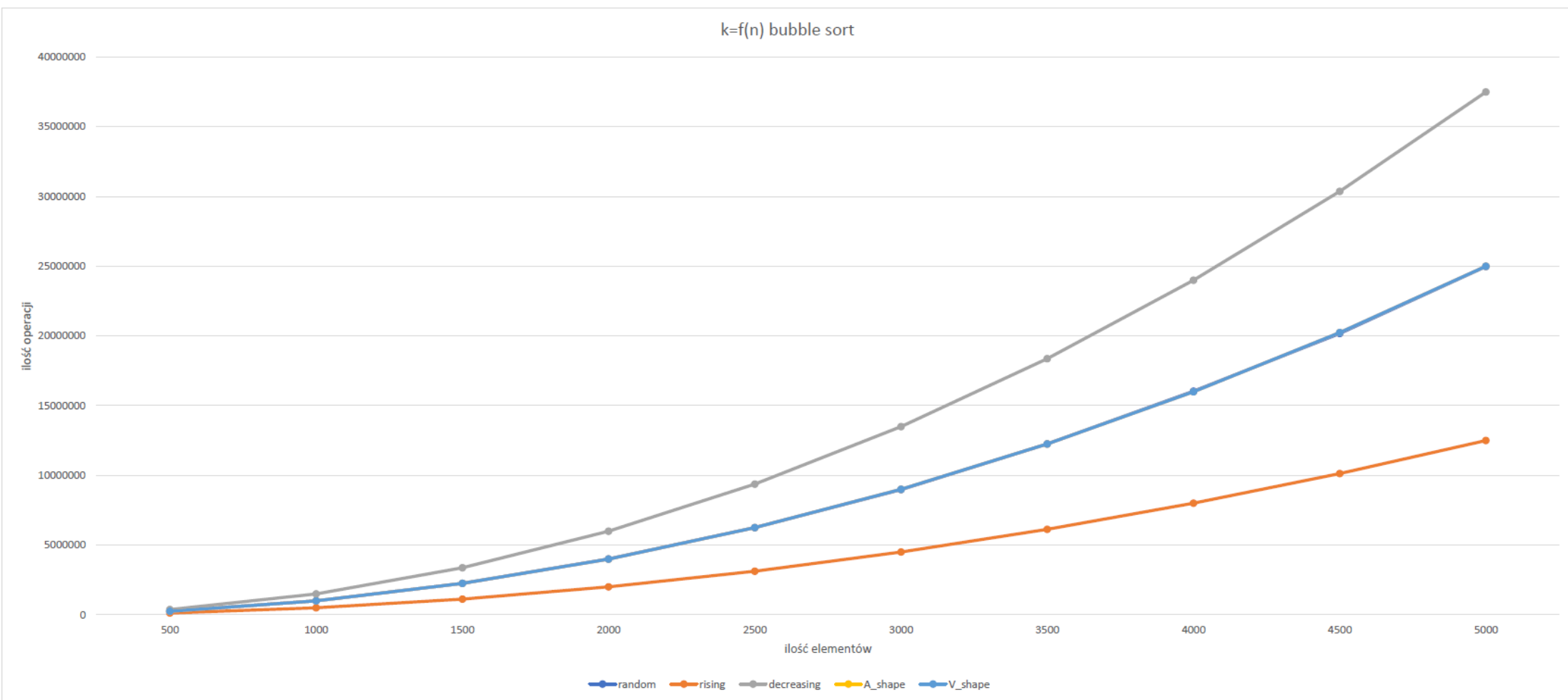
3c) Heap sort.



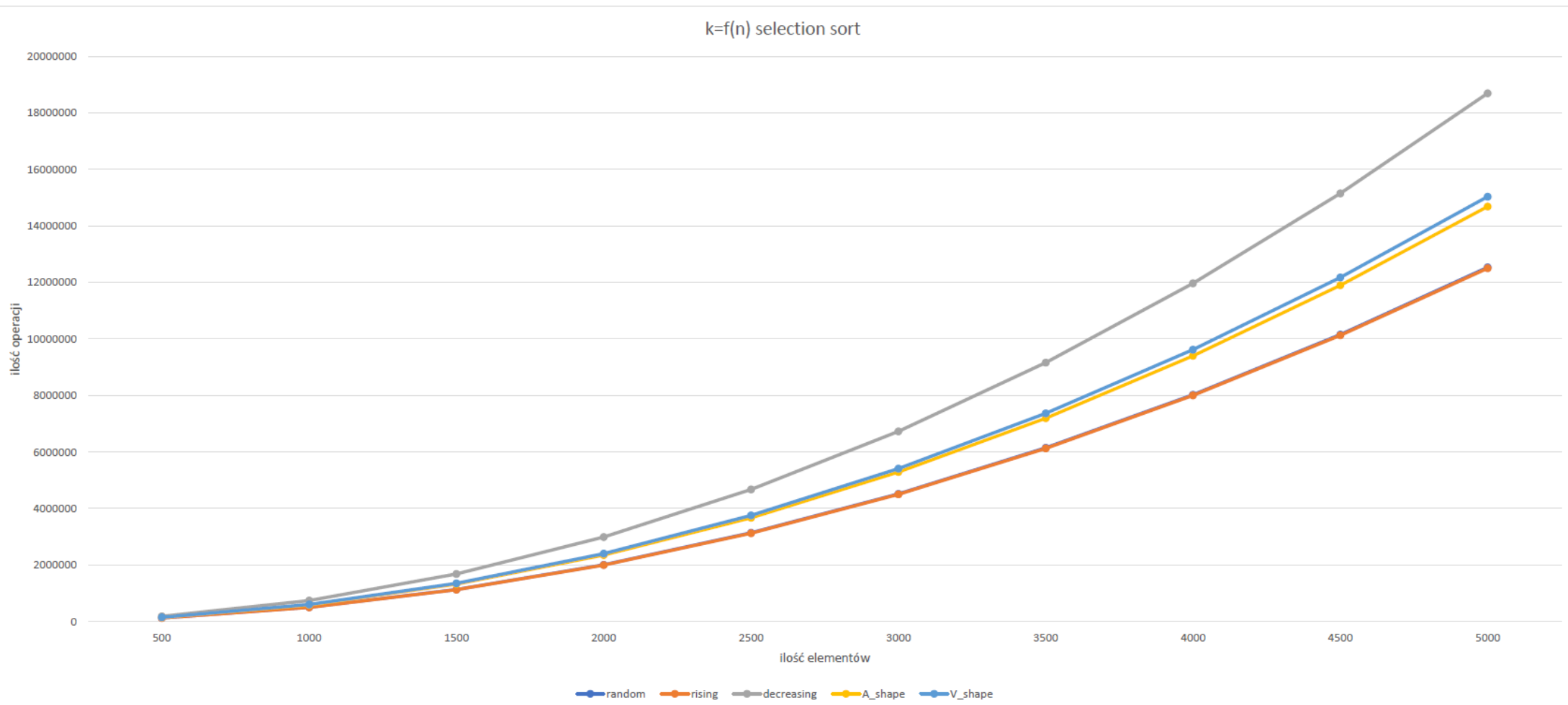
3d) Insertion sort. Wykres pokrywa się dla danych A kształtnych i V kształtnych. Na drugim wykresie przedstawiono tylko dan rosnące dla których zauważyć można, że liczba operacji < 2n



3e) Bubble sort. Różnica dla danych losowych, A kształtnych i V kształtnych jest w tej skali niezauważalna.



3f) Selection sort.



4) Złożoności obliczeniowe.

4a) Quick sort.

Przypadek optymistyczny: $O(n \log n)$

Przypadek pesymistyczny: $O(n^2)$

Przypadek średni: $O(n \log n)$

Komentarz: Jak potwierdzają wykresy quick sort najgorzej radzi sobie z danymi rosnącymi i malejącymi,

Wymaga to wykonania dużo większej liczby operacji, co niesie za sobą dłuższy czas wykonywania.

W tych przypadkach na krzywej wykresu zauważyć można parabolę, co oznacza złożoność kwadratową.

4b) Merge sort.

Przypadek optymistyczny: $O(n \log n)$

Przypadek pesymistyczny: $O(n \log n)$

Przypadek średni: $O(n \log n)$

Komentarz: Ilość operacji w każdym przypadku jest zbliżona, co sprawia,

że złożoność w każdym przypadku jest logarytmiczna.

4c) Heap sort.

Przypadek optymistyczny: $O(n \log n)$

Przypadek pesymistyczny: $O(n \log n)$

Przypadek średni: $O(n \log n)$

Komentarz: Sytuacja analogiczna do merge sort.

4d) Insertion sort

Przypadek optymistyczny: $O(n)$

Przypadek pesymistyczny: $O(n^2)$

Przypadek średni: $O(n^2)$

Komentarz: Jest to algorytm, który najlepiej radzi sobie z danymi, które już są posortowane. Złożoność w tym przypadku jest liniowa, a liczba operacji nie przekracza $2n$.

4e) Bubble sort.

Przypadek optymistyczny: $O(n^2)$

Przypadek pesymistyczny: $O(n^2)$

Przypadek średni: $O(n^2)$

Komentarz: Liczba operacji, które wykonuje algorytm jest tym mniejsza, czym większa jest liczba elementów, które są już we właściwej kolejności. Pomimo to złożoność w każdym przypadku jest kwadratowa.

4f) Selection sort

Przypadek optymistyczny: $O(n^2)$

Przypadek pesymistyczny: $O(n^2)$

Przypadek średni: $O(n^2)$

Komentarz: Sytuacja analogiczna do bubble sort.

