

Sprawozdanie

PAMSI lab6

Damian Jachowicz

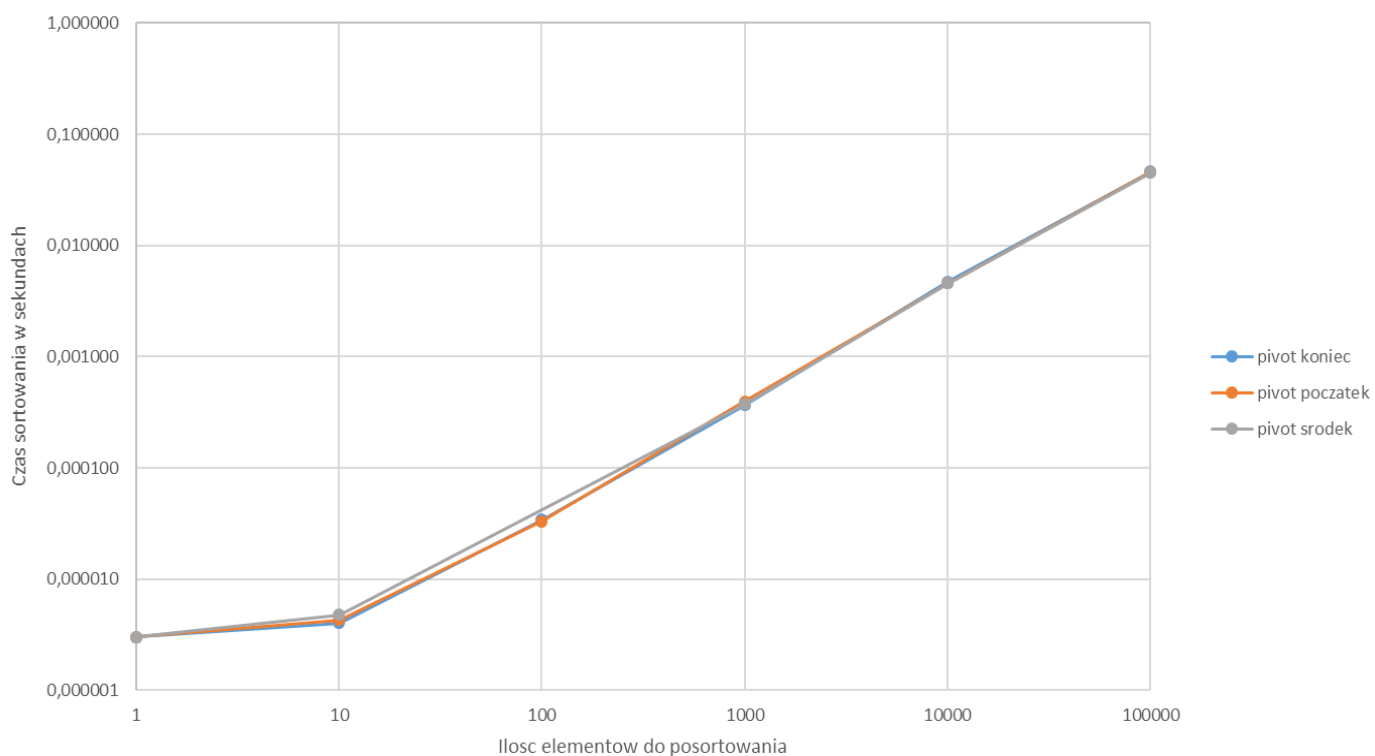
226238

czw. 17⁰⁵

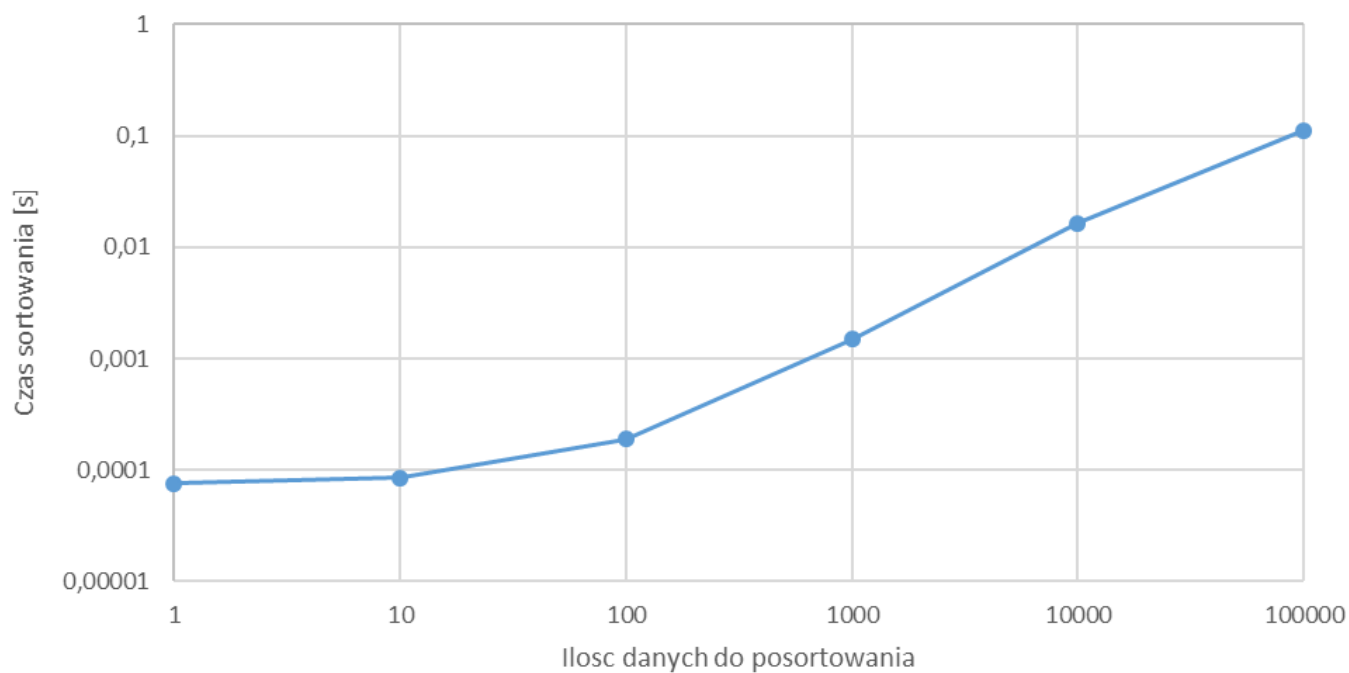
Quicksort vs heapsort

Zaimplementowano algorytm sortowania kopcowego. Sortowano tablicę na 3 sposoby – z losowym ułożeniem elementów, z malejącym ułożeniem elementów i z rosnącym ułożeniem elementów. Zmierzono średnie czasy wykonywania algorytmu. Dane zamieszczono na wykresie. Dodatkowo zamieszczono wykresy z sortowania szybkiego dla porównania.

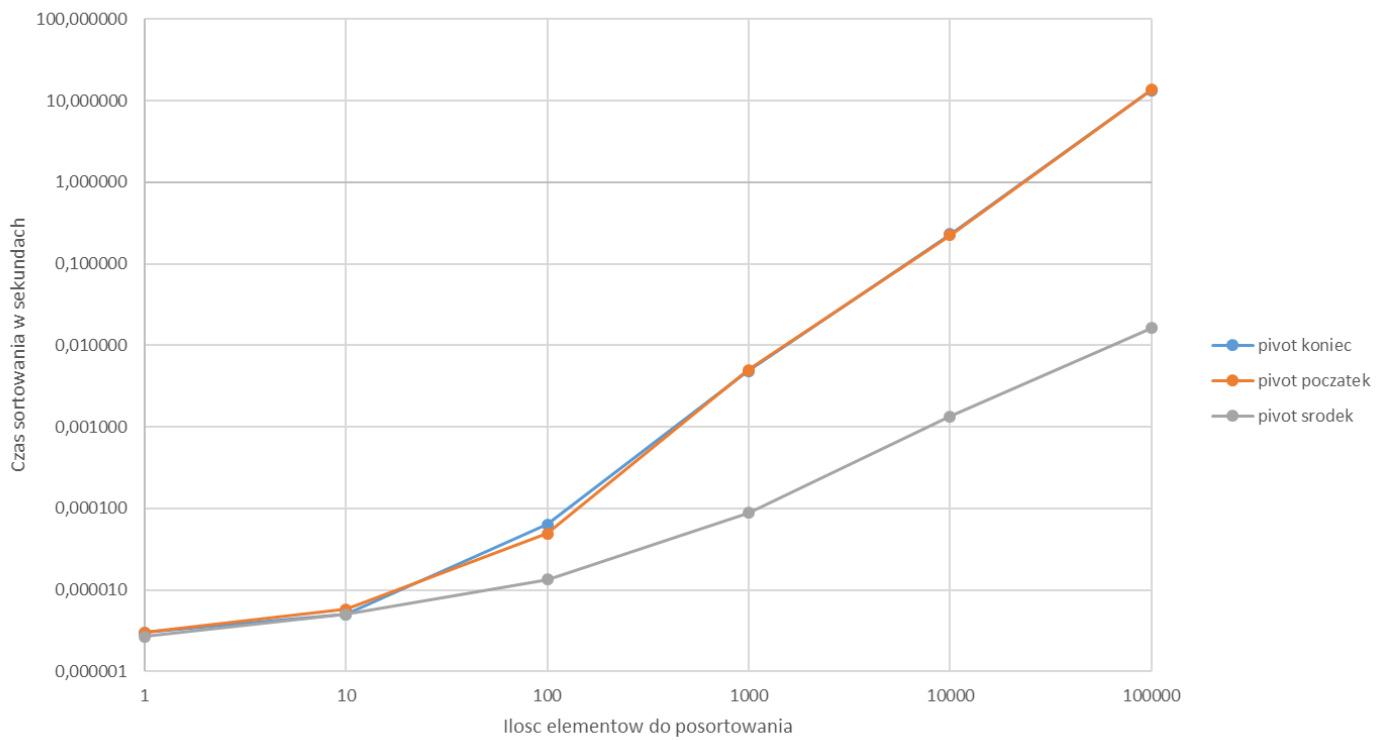
Tablica wypełniona losowymi elementami w losowej kolejności



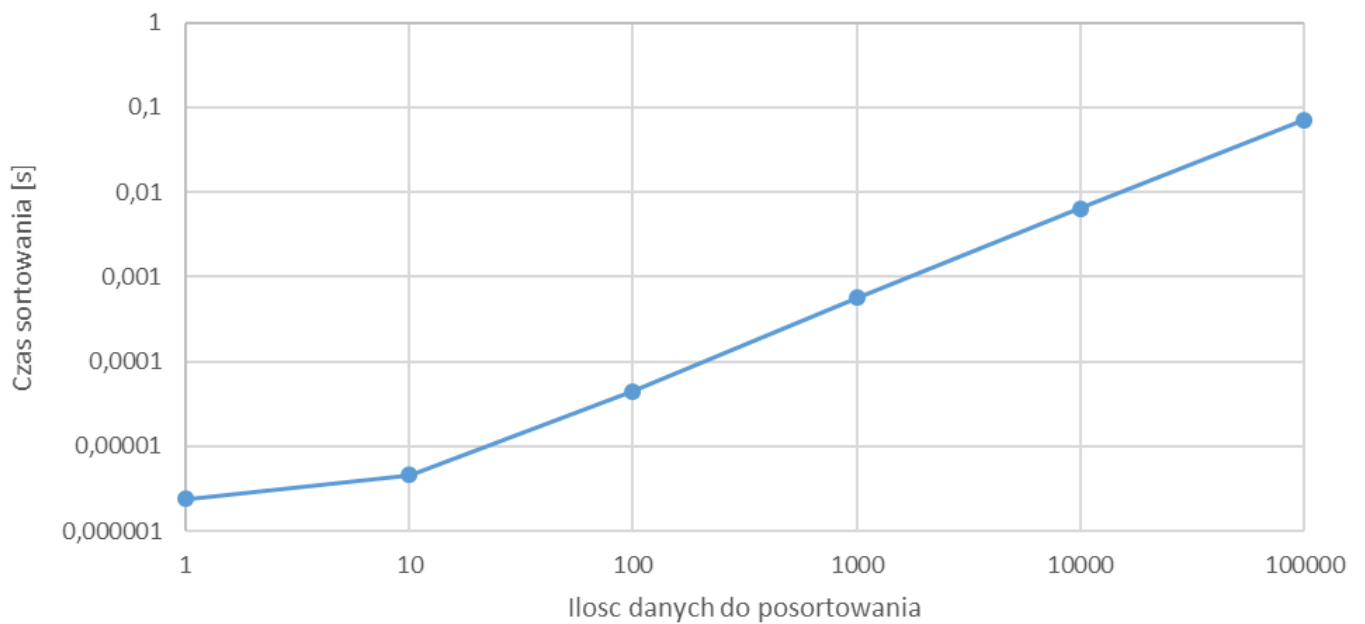
Sortowanie kopcowe, dane w kolejności losowej



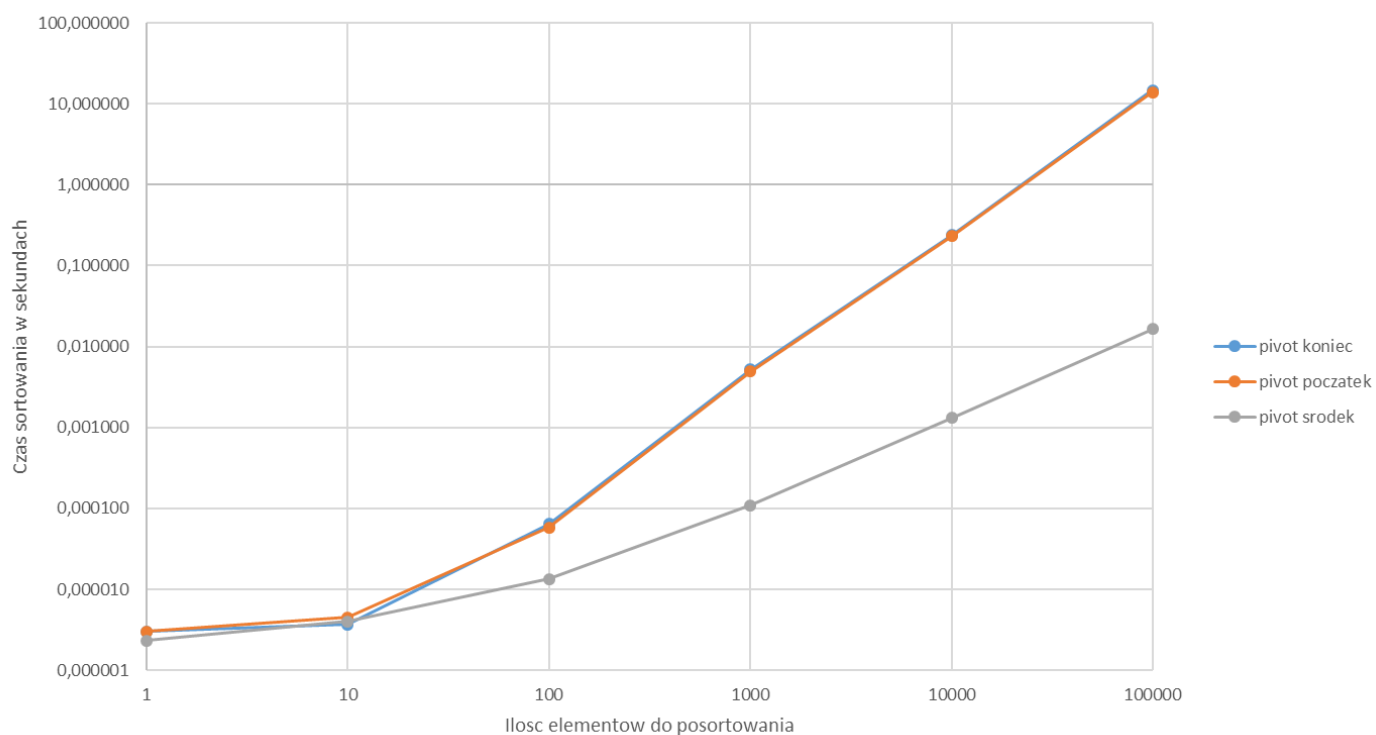
Tablica posortowana malejaco



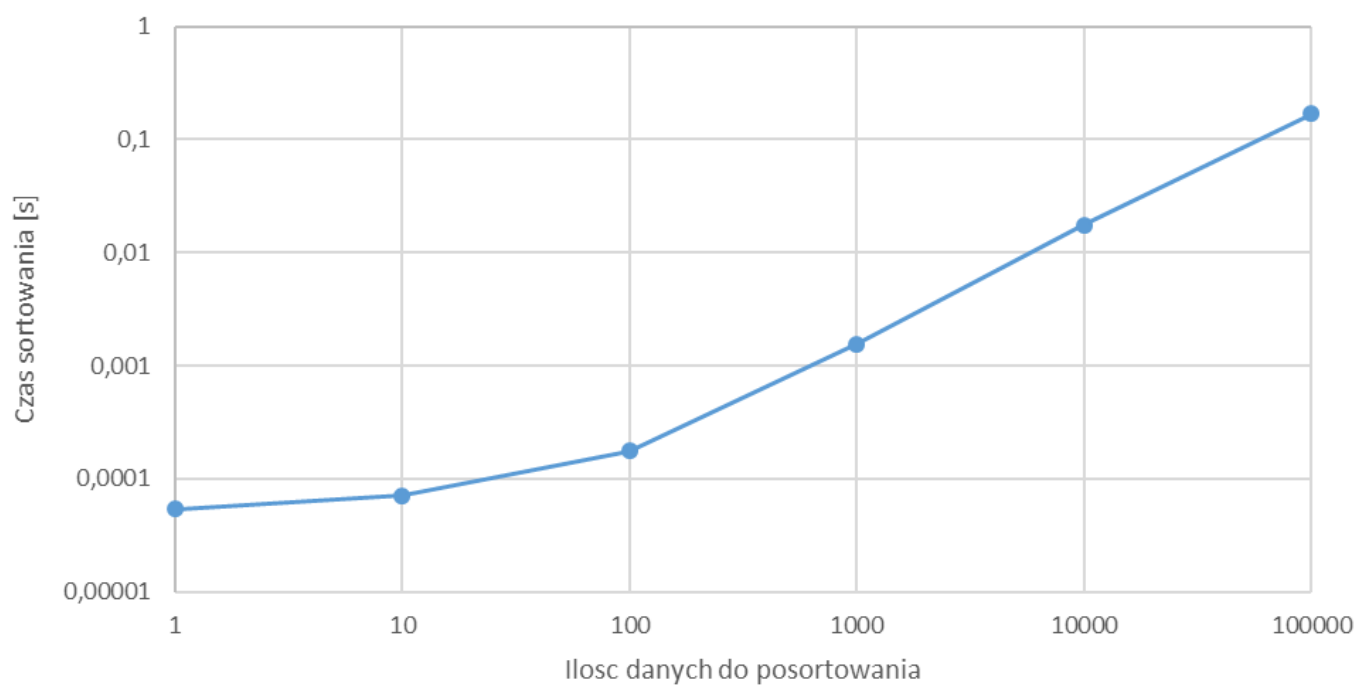
Sortowanie kopcowe, dane posortowane malejaco



Tablica posortowana rosnaco



Sortowanie kopcowe, dane posortowane rosnaco



Losowe ułożenie elementów:

Dla małej ilości danych szybszy jest algorytm sortowania szybkiego o niemal półtora rzędu wielkości. Dla większej ilości danych (od 100000) różnica ta znacznie się zmniejsza jednak nadal szybsze jest sortowanie szybkie.

Przedstawione wnioski takie same dla każdego ustawienia pivotu.

Malejące ułożenie elementów:

Dla ustawienia pivotu na końcu lub początku w sortowaniu szybkim algorytm sortowania kopcowego jest znacznie szybszy. Dla pivotu po środku niewiele szybszy okazuje się algorytm sortowania szybkiego.

Rosnące ułożenie elementów:

Dla ustawienia pivotu na końcu lub początku w sortowaniu szybkim przy małych ilościach danych algorytm sortowania kopcowego jest wolniejszy jednak dla większej ilości danych (już od 1000) znacznie szybciej wypada sortowanie kopcowe. Dla ustawienia pivotu po środku w sortowaniu szybkim dla każdej ilości danych operacje szybciej wykonuje sortowanie szybkie niż sortowanie kopcowe.

Wnioski:

Sortowanie kopcowe działa szybciej od sortowania szybkiego tylko dla tablic posortowanych malejąco lub rosnąco z pivotem umieszczonym na początku lub końcu (przypadki pesymistyczne). Dla ustawienia pivotu po środku zawsze szybciej wypadało sortowanie szybkie.

Sortowanie kopcowe osiąga cały czas mniej więcej podobne czasy bez względu na rozmieszczenie elementów w tablicy.

Złożoność czasowa sortowania kopcowego to około $O(n \cdot \log(n))$.