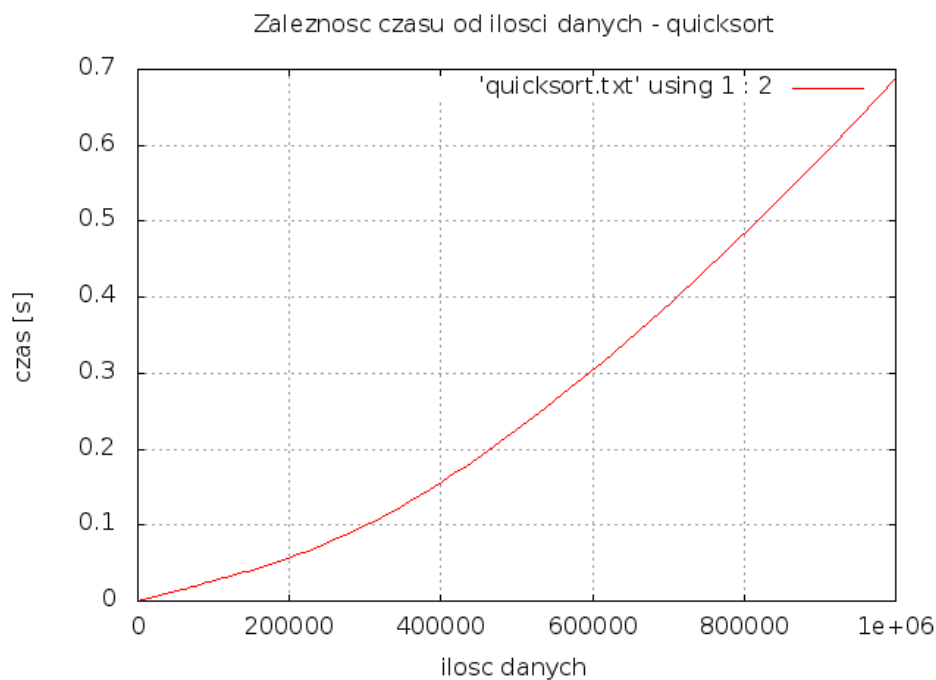


Roznice w czasie realizacji algorytmów sortowania

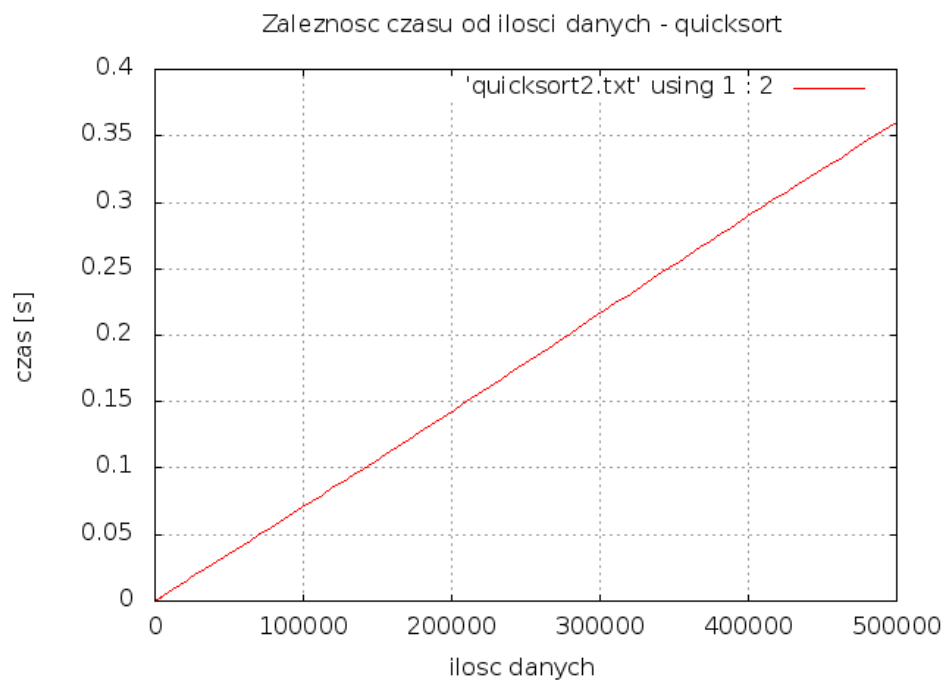
Arkadiusz Cyktor 200367

23 marca 2014

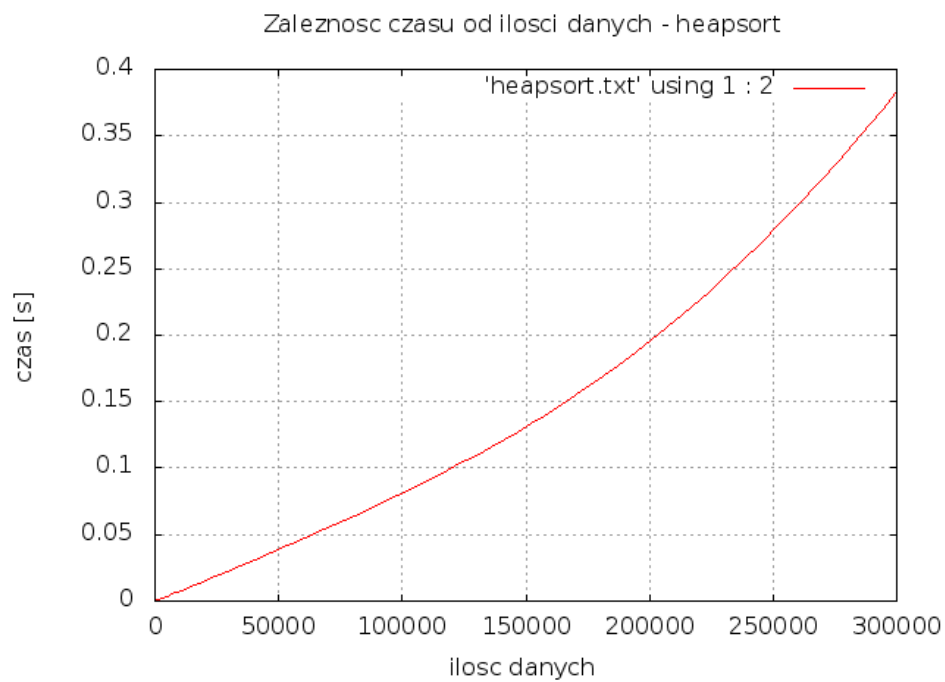
1. Poniższy wykres przedstawia zależność czasu potrzebnego na wykonanie algorytmu **quicksort**, jak widać rośnie ona wykładniczo.



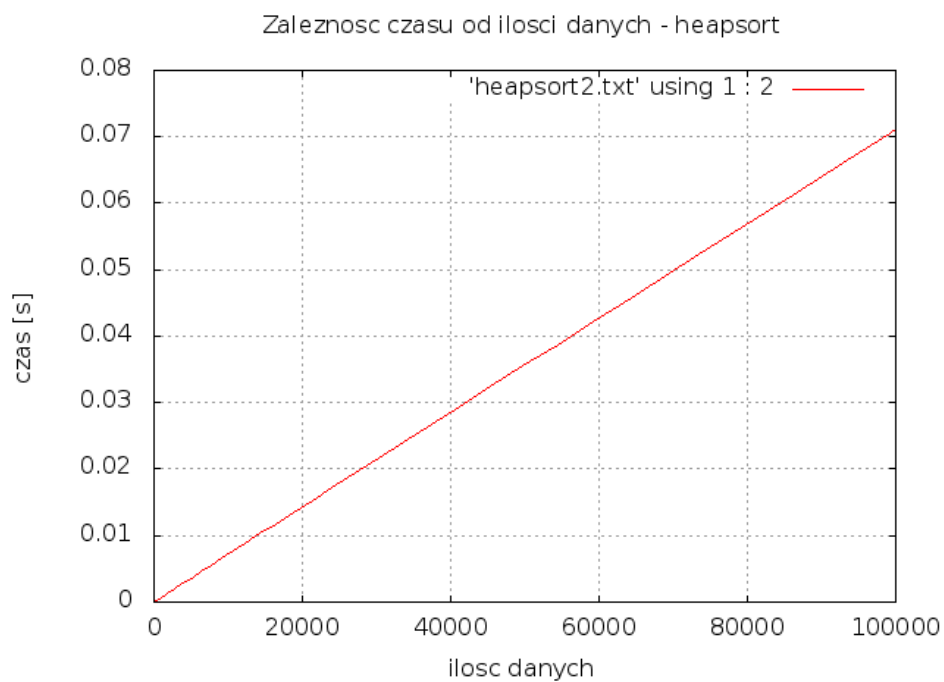
Teoretyczna średnia złożoność obliczeniowa tego algorytmu jest bliska złożoności w przypadku optymistycznym (wtedy gdy za każdym razem udaje się wybrać medianę z sortowanego fragmentu tablicy) i wynosi $n \log n$. Wyniki moich testów, bardziej odpowiadają przypadkowi pesymistycznemu, w którym występuje złożoność kwadratowa. Wynika to z faktu, że przy liczbie powtórzeń algorytmu większej niż 1, sortowana będzie posortowana już tablica - przypadek pesymistyczny. Po zamianie liczby powtórzeń na 1 złożoność zmienia się według zależności $n \log n$. W pewnym stopniu obrazuje to poniższy wykres.



2. Poniższy wykres przedstawia zależność czasu potrzebnego na wykonanie algorytmu **heapsort**.



Jak, może niezbyt dobrze, widac na wykresie - złożoność obliczeniowa zmienia się według zależności $n \log n$, co odpowiada pesymistycznemu przypadkowi dla tego algorytmu (każde wstawienie nowego elementu na stos wymaga tylu przesunień, ile jest węzłów na ścieżce od miejsca wstawienia do korzenia drzewa). Po zmianie liczby powtórzeń na 1 otrzymałem zależność liniową - n , więc testowany przypadek był optymistyczny (dodanie nowego elementu nie naruszało struktury kopca).



3. Poniższy wykres przedstawia zależność czasu potrzebnego na wykonanie algorytmu **mergesort**. Algorytm ten charakteryzuje się taką samą złożonością obliczeniową dla każdego przypadku - wynosi ona $n \log n$. Obrazuje to poniższy wykres (zależność $n \log n$ łatwo pomylić z wykładniczą, zwłaszcza w przypadku takim jak ten - gdy z oczywistych względów oś y nie przyjmuje wartości ujemnych).

