

Sprawozdanie z laboratorium

Sortowanie

Adam Dąbrowski 184208

May 6, 2014

0.1 Wprowadzenie

Celem ćwiczenia było zapoznanie się z różnymi typami sortowania. Ja zaimplementowałem heapsort i quicksort. Oba te algorytmy sortują w miejscu to znaczy nie wymagają żadnej dodatkowej struktury danych.

0.2 Realizacja

Program sprawdzam kolejno dla wielkości problemu 10, 100 ,1000,10 000, 100 000. Sortowanie przeprowadzam 50 razy, przy tej liczbie przy prawie każdym uruchomieniu dostaję podobne wyniki, a następnie liczę z tego średni czas. Dobrym punktem odniesienia przy analizie algorytmów sortowania jest podstawowe sortowanie bąbelkowe, ponieważ po kolei przestawia w tablicy wszystkie elementy- dlatego też swoją pracę zacząłem od napisanie tego typu sortowania.

0.3 HEAPSORT

Złożoność obliczeniowa

Funkcja heapSort - $O(n \lg n)$

Funkcja buildMaxHeap - $O(n)$

Funkcja maxHeapify każde z $n-1$ wywołań - $O(\lg n)$

HEAPSORT	
ilosc elementow	czas - nano sekundy
10	4099
100	41165
1000	301103
10000	3905768
100000	5820387

0.4 QUICKSORT

Złożoność obliczeniowa w wypadku funkcji quicksort bardzo mocno zależy od danych wejściowych

W najgorszym wypadku-dane już posortowane $O(n^2)$ Jednak zazwyczaj jest to $O(n \lg n)$

QUICKSORT- przypadek średni	
ilosc elementow	czas - nano sekundy
10	1725
100	26031
1000	144574
10000	1834571
100000	27507827
1000000	27507827
10000000	333791008

QUICKSORT- przypadek najgorszy	
ilosc elementow	czas - nano sekundy
10	5035
100	334191
1000	7576229
10000	759267000

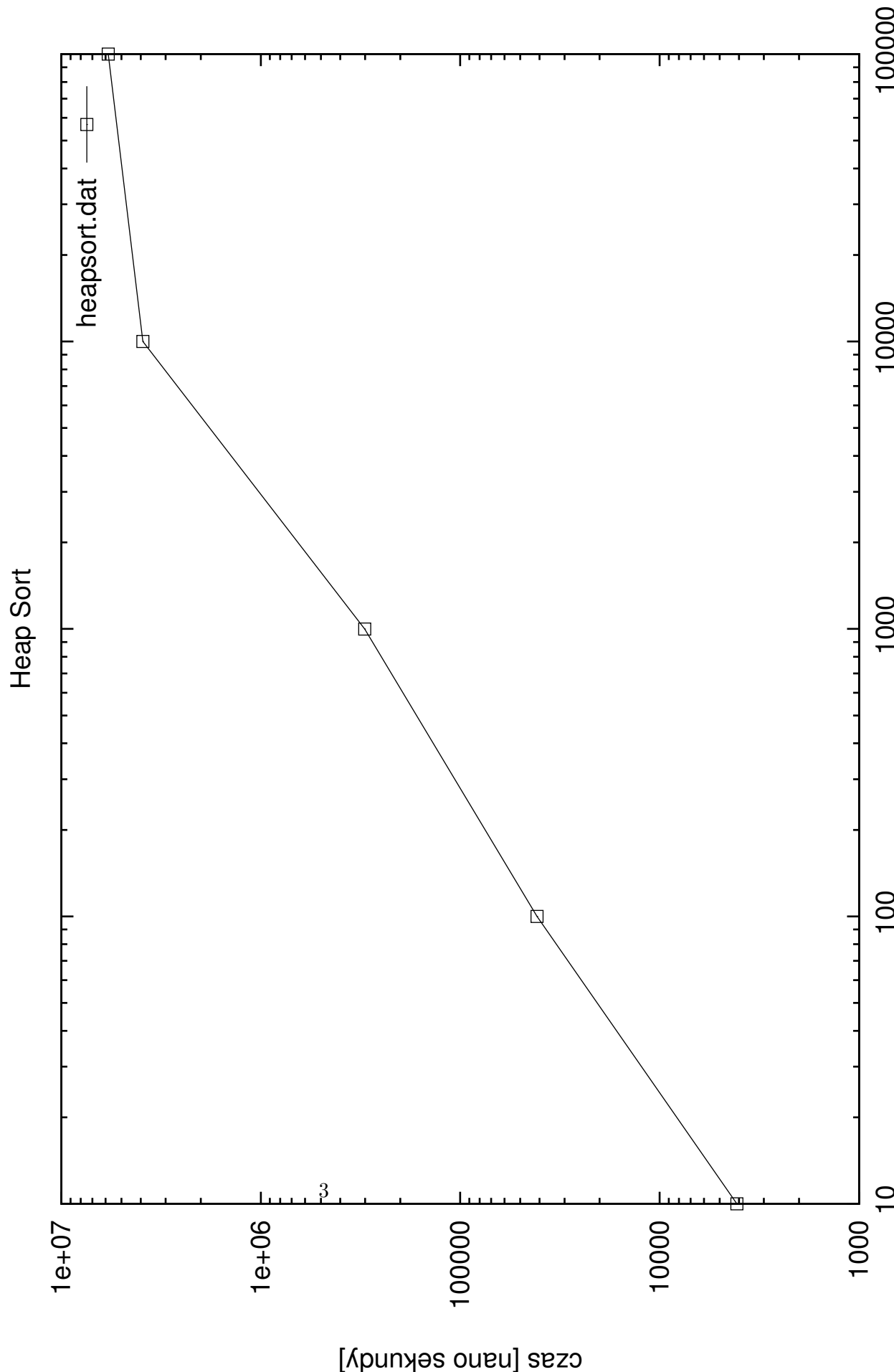
0.5 BUBBLESORT

Złożoność obliczeniowa - $O(n^2)$

BUBBLESORT	
ilosc elementow	czas - nano sekundy
10	3878
100	186381
1000	8346056
10000	769738534

0.6 Wnioski

Quicksort w najgorszym wypadku podobnie jak bubblesort ma złożoność obliczeniową $O(n^2)$, jednak dla przypadku średniego jest to $O(n \lg n)$. Potwierdzają to otrzymane czasy realizacji gdzie quicksort(przypadek średni) byłem w stanie wywołać dla $n = 10^7$ a bubblesort już tylko dla $n = 10^4$. Dla przypadku najgorszego quicksort otrzymane czasy realizacji były gorsze niż dla sortowania bąbelkowego. Czasy realizacji dla algorytmu heapsort zgodnie z założeniami teoretycznymi są bardzo zbliżone do tych quicksort'a dla przypadku średniego.



Quick Sort

