

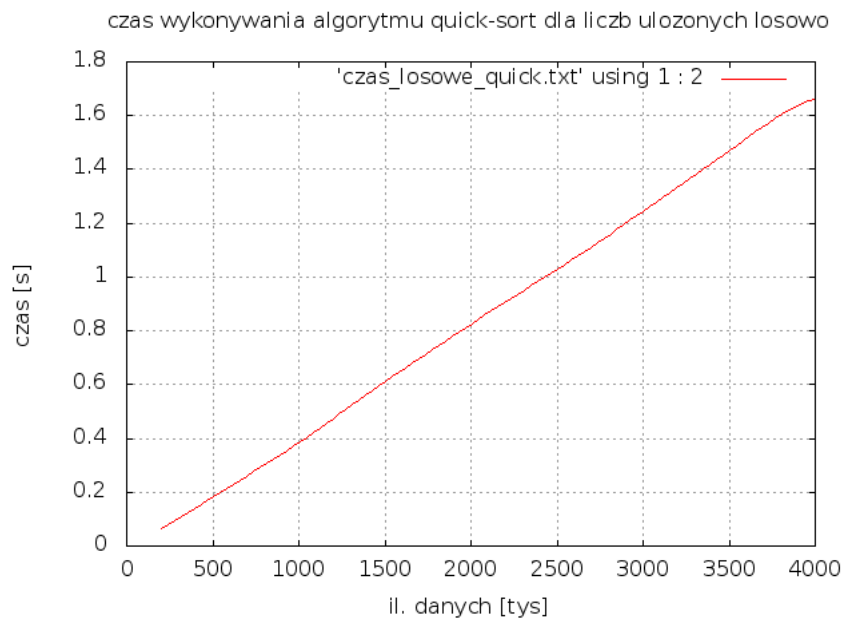
czas obsługi: stosu, listy

Tomasz Piotrowski 200524

7 kwietnia 2014

Sprawozdanie z czasu działania algorytmów obsługi stosu oraz listy.

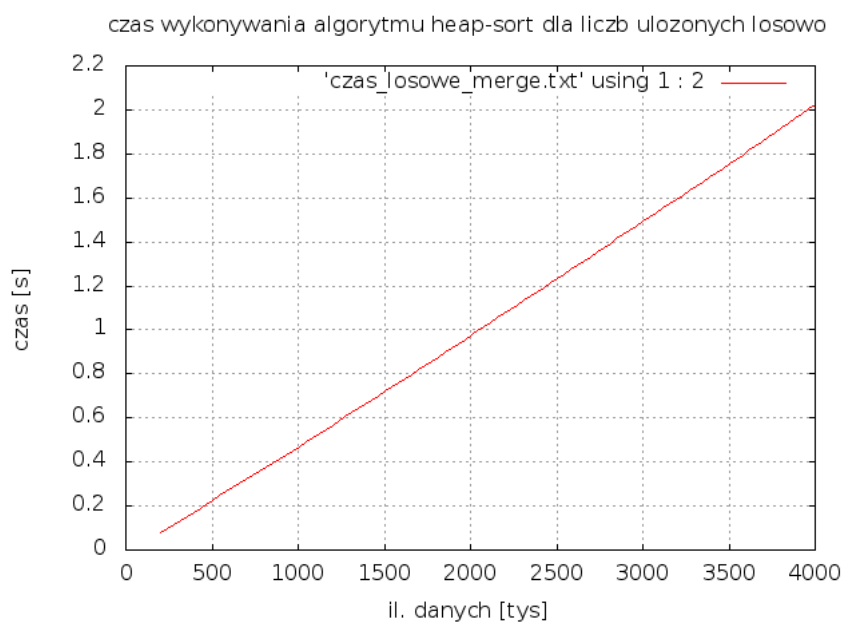
1.Sortowanie za pomoca algorytmu quicksort losowo wygenerowanych liczb. Pliki z wynikami dołączone do sprawozdania. Z wykresu można wywnioskować że złożoność działania algorytmu jest liniowa.



Rysunek 1:

il[tys]	czas pól [s]
200	0.066
400	0.148
600	0.218
800	0.298
1000	0.374
1200	0.464
1400	0.548
1600	0.696
1800	0.754
2000	0.81
2200	0.922
2400	0.982
2600	1.06
2800	1.142
3000	1.236
3200	1.354
3400	1.414
3600	1.508
3800	1.634
4000	1.66

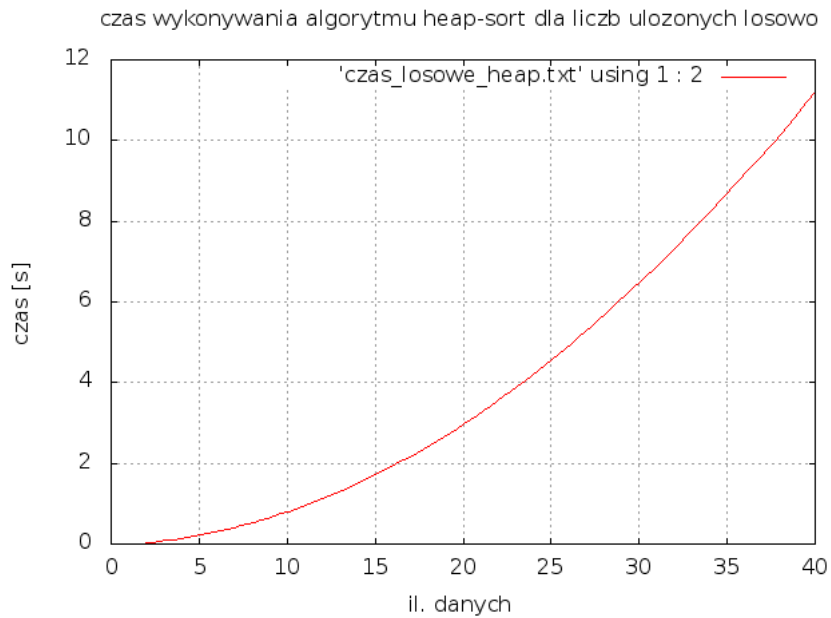
1.Sortowanie za pomocą algorytmu sacalania losowo wygenerowanych liczb. Pliki z wynikami dołączone do sprawozdania. Z wykresu można wywnioskować że złożoność działania algorytmu jest liniowa.



il[tys]	czas pół [s]
200	0.082
400	0.176
600	0.272
800	0.368
1000	0.462
1200	0.566
1400	0.67
1600	0.768
1800	0.862
2000	0.976
2200	1.078
2400	1.184
2600	1.278
2800	1.39
3000	1.49
3200	1.594
3400	1.704
3600	1.806
3800	1.908
4000	2.026

Rysunek 2:

1.Sortowanie za pomocą algorytmu kopcowania losowo wygenerowanych liczb. Pliki z wynikami dołączone do sprawozdania. Z wykresu można wywnioskować że złożoność działania algorytmu jest wykładnicza.



il[tys]	czas pół [s]
2	0.03
4	0.116
6	0.258
8	0.46
10	0.71
12	1.02
14	1.388
16	1.826
18	2.284
20	2.806
22	3.432
24	4.06
26	4.794
28	5.548
30	6.418
32	7.276
34	8.172
36	9.094
38	10.104
40	11.208

Rysunek 3:

Testy wykazały że w przypadku losowo wygenerowanych liczb. Algorytm sortowania szybkiego oraz przez scalanie mają podobną złożoność obliczeniową. Jednak Algorytm sortowania szybkiego wykonał operację o 0.4s szybciej niż algorytm sortowania przez scalanie. Algorytm sortowania kopcowego nadaje się do sortowania małych ilości danych. Przewagą algorytmu sortowania kopcowego jest jego małe zapotrzebowanie na pamięć. W przypadku losowo ułożonych liczb algorytm sortowania szybkiego jest najszybszy.

