

MERGE SORT

DZIAŁANIE ALGORYTMU

MergeSort opiera się na rekurencyjnym dzieleniu tablicy na połowki, do czasu aż zostaną jedynie pojedyncze elementy, następnie porównuje się coraz większe kawałki połówek (oddzielnie) tablicy w razie potrzeby zamieniając kolejność i scalając je. Po tym etapie mamy 2 posortowane w swoim zakresie części tablicy – scalamy je do ostatecznego rezultatu algorytmem porównującym 2 wybrane pola z różnych połówek tablicy: jeśli element A jest większy od B, to kopiujemy B do tablicy wynikowej i przesuwamy pole w podtablicy zawierającej B. Minusem algorytmu jest potrzeba zagospodarowania pamięci dla tablicy wynikowej, plusem jest złożoność niezależna od danych – jak widać poniżej, wyniki nie różnią się specjalnie dla żadnej modyfikacji wejściowych danych w obrębie jednej liczności. Złożoność ta wynosi $T(n) = 2n \cdot \log(n)$ i należy do klasy $O(n \log n)$.

DLA PSEUDOLOSOWYCH

1000

real	0m0.019s
user	0m0.000s
sys	0m0.004s

2000

real	0m0.020s
user	0m0.012s
sys	0m0.000s

4000

real	0m0.021s
user	0m0.008s
sys	0m0.004s

8000

real	0m0.107s
user	0m0.008s
sys	0m0.004s

16000

real	0m0.153s
user	0m0.024s
sys	0m0.004s

32000

real	0m0.342s
user	0m0.024s
sys	0m0.020s

64000

real	0m0.776s
user	0m0.072s
sys	0m0.016s

128000

real	0m1.531s
user	0m0.140s
sys	0m0.024s

DLA SORTED

4000

real	0m0.020s
user	0m0.004s
sys	0m0.004s

32000

real	0m0.405s
user	0m0.028s
sys	0m0.008s

128000

real	0m1.371s
user	0m0.116s
sys	0m0.036s

DLA SORTED REVERSE

4000

real	0m0.022s
user	0m0.000s
sys	0m0.008s

32000

real	0m0.064s
user	0m0.024s
sys	0m0.012s

128000

real	0m1.364s
user	0m0.104s
sys	0m0.032s

DLA SORTED Z LOSOWYM ELEMENTEM NA POEZATKU

4000

real	0m0.023s
user	0m0.004s
sys	0m0.004s

32000

real	0m0.183s
user	0m0.024s
sys	0m0.012s

128000

real	0m1.307s
user	0m0.108s
sys	0m0.044s

DLA SORTED Z LOSOWYM ELEMENTEM NA KONCU

4000

real	0m0.029s
user	0m0.004s
sys	0m0.004s

32000

real	0m0.166s
user	0m0.024s
sys	0m0.016s

128000

real	0m1.497s
user	0m0.104s
sys	0m0.040s