Raport

Po zastosowaniu kilku algorytmów sortowania w kilku wariantach następujące są wnioski:

(Na tabelach poniżej widać średni czas dla danej ilości elementów

wygenerowanej w dany sposób.)

Selection Sort:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ilość/Typ | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 100000 |
| Losowo | 0.2420 | 0.9770 | 2.1440 | 3.8000 | 5.7770 | 8.1010 | 10.706 | 13.906 | 17.333 | 20.463 |
| Rosnąco | 0.0990 | 0.3910 | 0.8760 | 1.5840 | 2.4480 | 3.5610 | 4.8360 | 6.4950 | 8.0790 | 9.8430 |
| Malejąco | 0.1030 | 0.4130 | 0.9470 | 1.6560 | 2.6470 | 3.7660 | 5.1400 | 6.6720 | 8.4010 | 10.425 |
| CZAS | | | | | | | | | | |

Selection Sort ma najwyższy średni czas dla elementów generowanych losowo i elementów generowanych rosnąco. W zamian ma niższy średni czas sortowania dla elementów malejących od “Insertion Sort”.

Insertion Sort:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 100000 |
| Losowo | 0.1160 | 0.4620 | 1.0530 | 1.8800 | 2.9070 | 4.2290 | 5.7970 | 7.4890 | 9.5960 | 11.777 |
| Rosnąco | 0.0840 | 0.3380 | 0.7550 | 1.4180 | 2.1800 | 3.1920 | 4.2680 | 5.6060 | 7.0060 | 8.8070 |
| Malejąco | 0.1210 | 0.4890 | 1.1030 | 2.0440 | 3.1750 | 4.5130 | 6.2120 | 7.8500 | 9.9790 | 12.813 |
| CZAS | | | | | | | | | | |

Insertion Sort jest lepszy w dwóch wariantach od swojego “Selection Sort”. Jest zdecydowanie lepszy dla elementów losowych i tylko trochę lepszy dla elementów generowanych rosnąco (różnica jednej sekundy). Dla liczb malejących jest już bardziej widoczna różnica co oznacza, że przy wyższych ilościach elementów do sortowania czas może się znacząco zwiększyć.

Heap Sort:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 100000 |
| Losowo | 0.0010 | 0.0040 | 0.0050 | 0.0080 | 0.0110 | 0.0120 | 0.0150 | 0.0160 | 0.0200 | 0.022000 |
| Rosnąco | 0.0010 | 0.0030 | 0.0070 | 0.0070 | 0.0090 | 0.0100 | 0.0130 | 0.0150 | 0.0170 | 0.023000 |
| Malejąco | 0.0010 | 0.0030 | 0.0050 | 0.0070 | 0.0090 | 0.0110 | 0.0130 | 0.0130 | 0.0160 | 0.019000 |
| CZAS | | | | | | | | | | |

Heap sort ma najniższe czasy w każdej z kategorii ze wszystkich tutaj pokazanych algorytmów, ale można go stosować tylko dla liczb. Różnica jest bardzo widoczna gdy zobaczymy wartości czasu przy stu tysiącach elementów.

