

Piotr Świątek

146 551

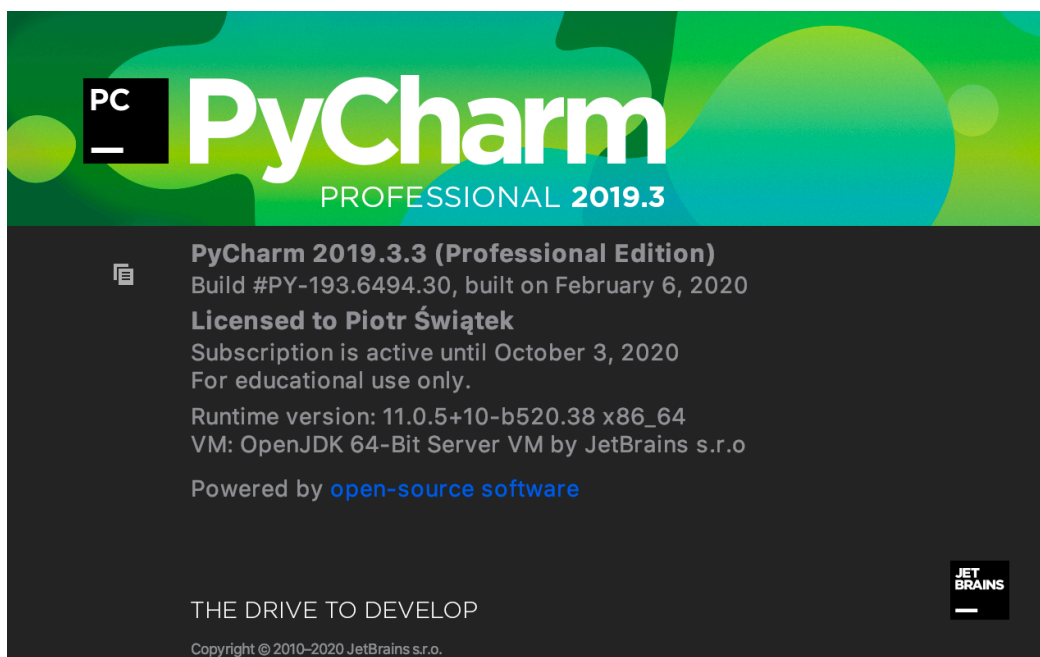
Grupa 4B

## Algorytmy sortowania

Benchmark hardware:

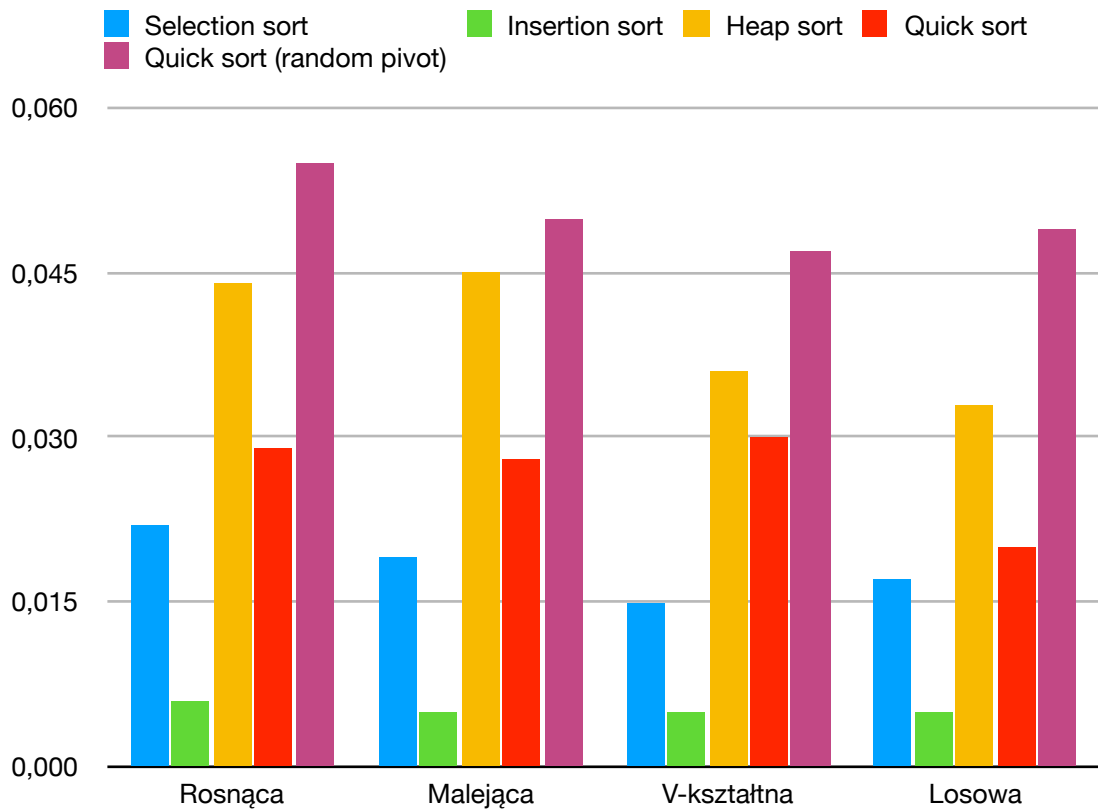


Software:



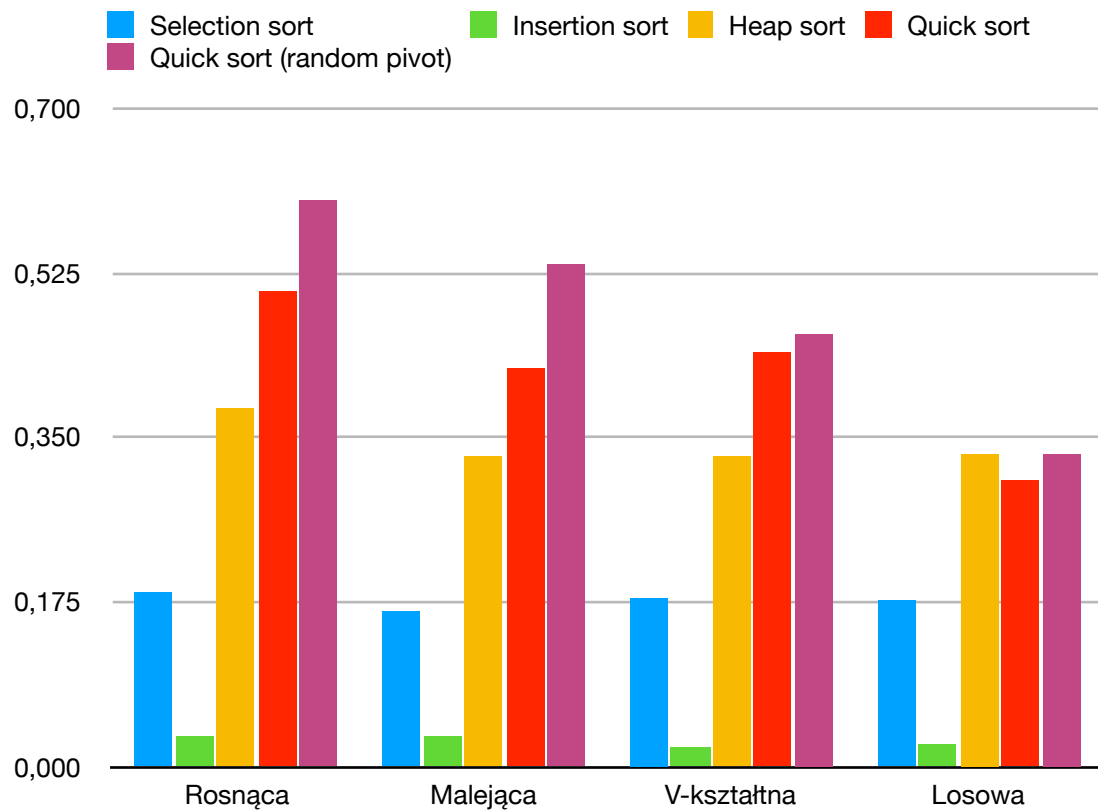
# 10 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	0.022 ms	0.006 ms	0.044 ms	0.029 ms	0.055 ms
<b>Tablica malejąca</b>	0.019 ms	0.005 ms	0.045 ms	0.028 ms	0.05 ms
<b>Tablica V-kształtna</b>	0.015 ms	0.005 ms	0.036 ms	0.03 ms	0.047 ms
<b>Tablica losowa</b>	0.017 ms	0.005 ms	0.033 ms	0.02 ms	0.049 ms



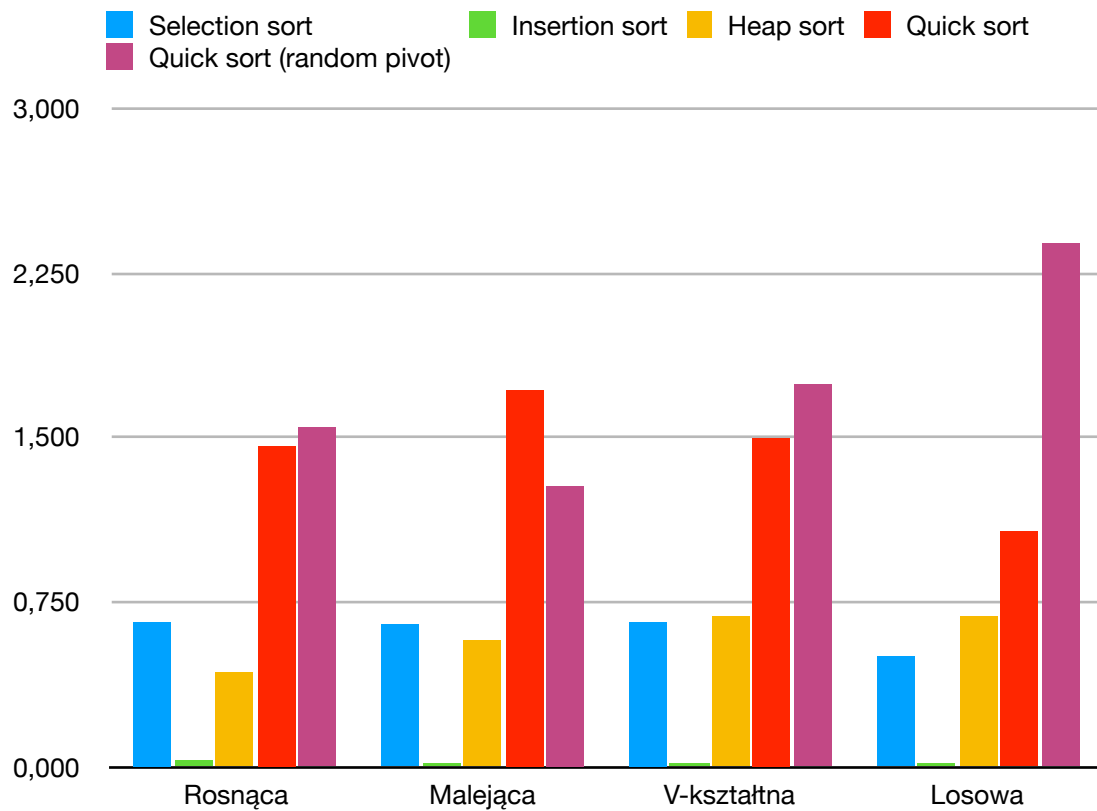
# 50 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	0.185 ms	0.031 ms	0.379 ms	0.505 ms	0.603 ms
<b>Tablica malejąca</b>	0.165 ms	0.033 ms	0.329 ms	0.423 ms	0.534 ms
<b>Tablica V-kształtna</b>	0.179 ms	0.019 ms	0.329 ms	0.441 ms	0.459 ms
<b>Tablica losowa</b>	0.176 ms	0.022 ms	0.333 ms	0.303 ms	0.333 ms



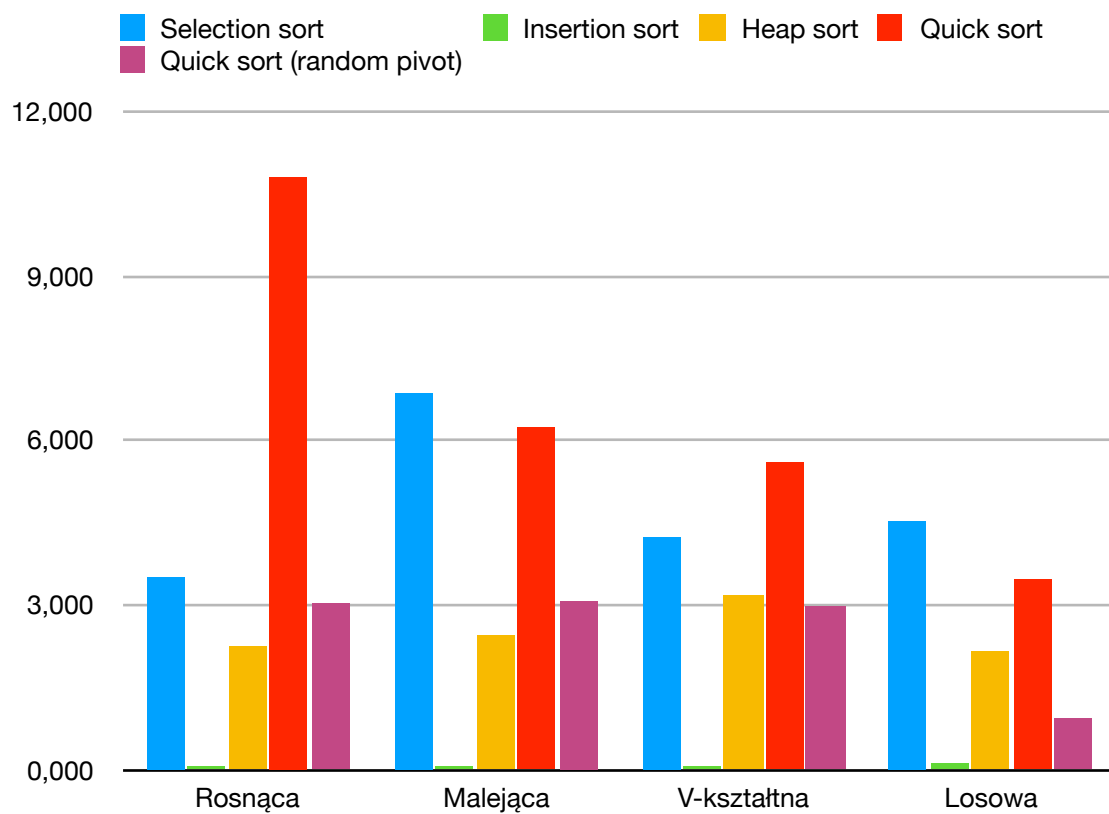
# 100 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	0.652 ms	0.02 ms	0.428 ms	1.455 ms	1.543 ms
<b>Tablica malejąca</b>	0.646 ms	0.018 ms	0.57 ms	1.716 ms	1.275 ms
<b>Tablica V-kształtna</b>	0.66 ms	0.017 ms	0.695 ms	1.497 ms	1.737 ms
<b>Tablica losowa</b>	0.499 ms	0.018 ms	0.677 ms	1.077 ms	2.388 ms



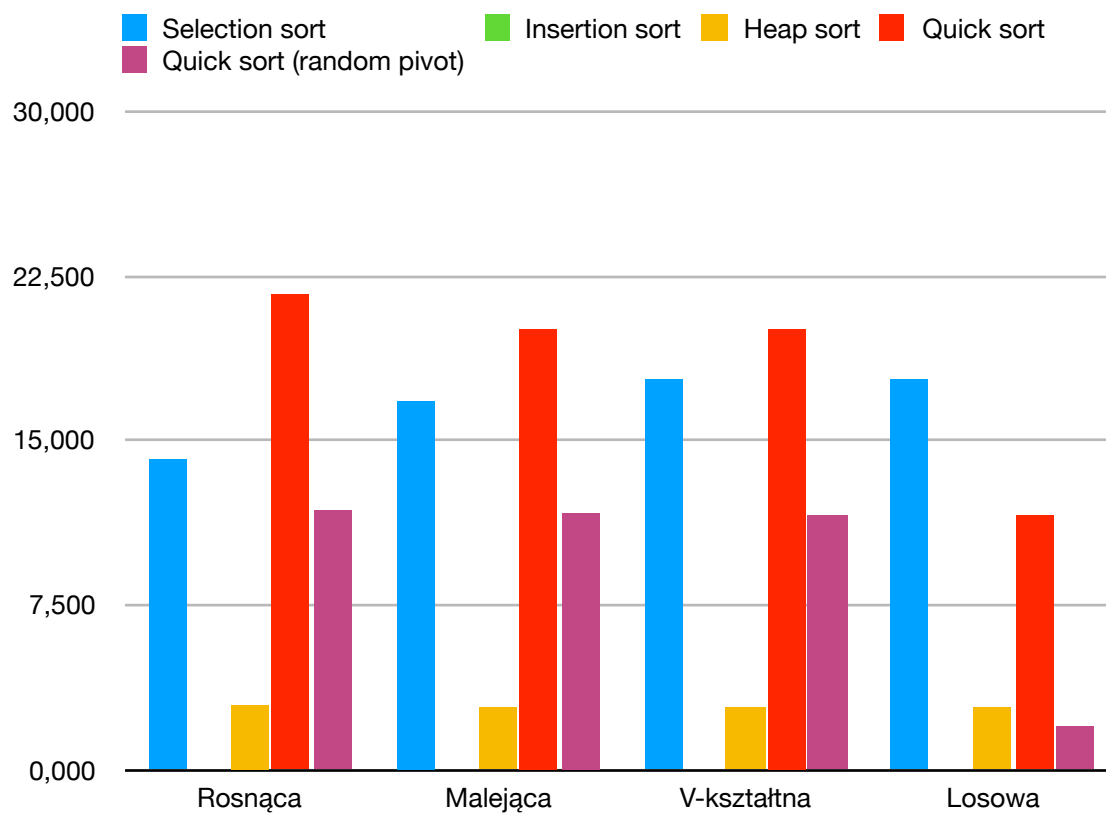
## 250 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	3.532 ms	0.077 ms	2.244 ms	10.781 ms	3.01 ms
<b>Tablica malejąca</b>	6.875 ms	0.073 ms	2.415 ms	6.222 ms	3.058 ms
<b>Tablica V-kształtna</b>	4.255 ms	0.075 ms	3.18 ms	5.595 ms	2.986 ms
<b>Tablica losowa</b>	4.547 ms	0.083 ms	2.135 ms	3.458 ms	0.924 ms



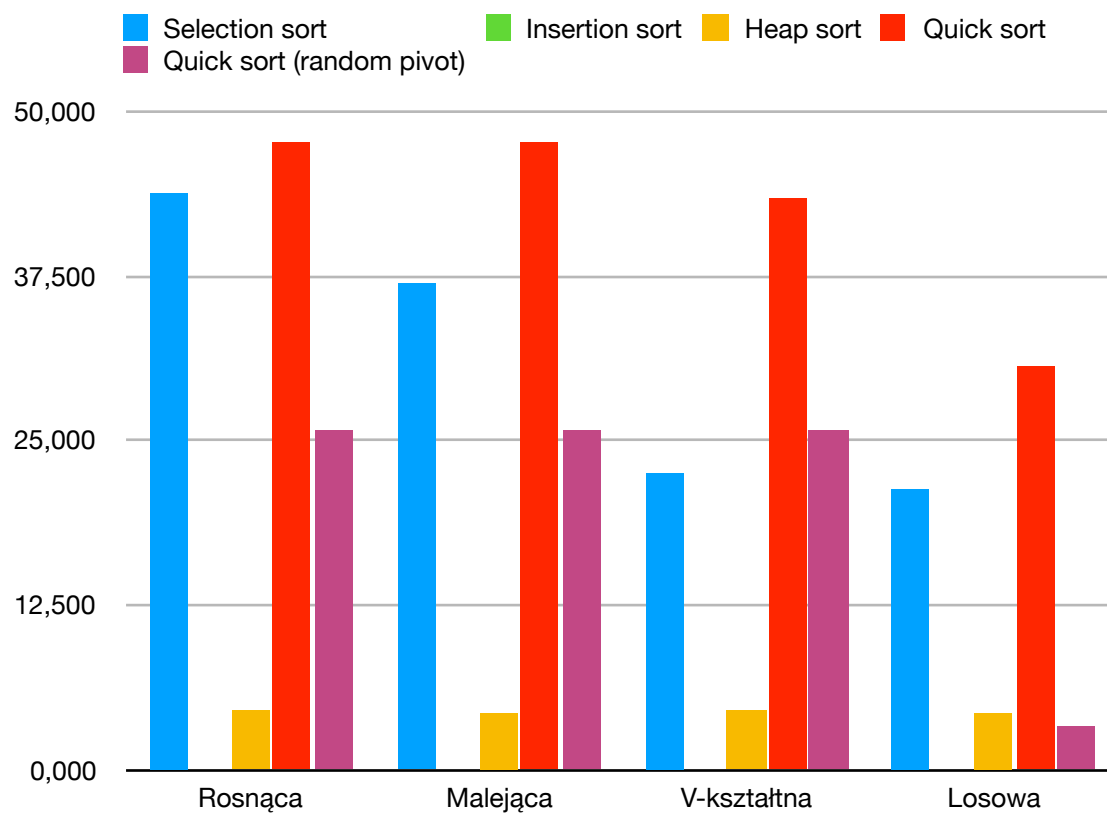
## 500 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	14.142 ms	0.098 ms	2.984 ms	21.686 ms	11.847 ms
<b>Tablica malejąca</b>	16.835 ms	0.095 ms	2.835 ms	20.024 ms	11.727 ms
<b>Tablica V-kształtna</b>	17.769 ms	0.095 ms	2.854 ms	20.081 ms	11.571 ms
<b>Tablica losowa</b>	17.124 ms	0.096 ms	2.77 ms	12.617 ms	1.94 ms



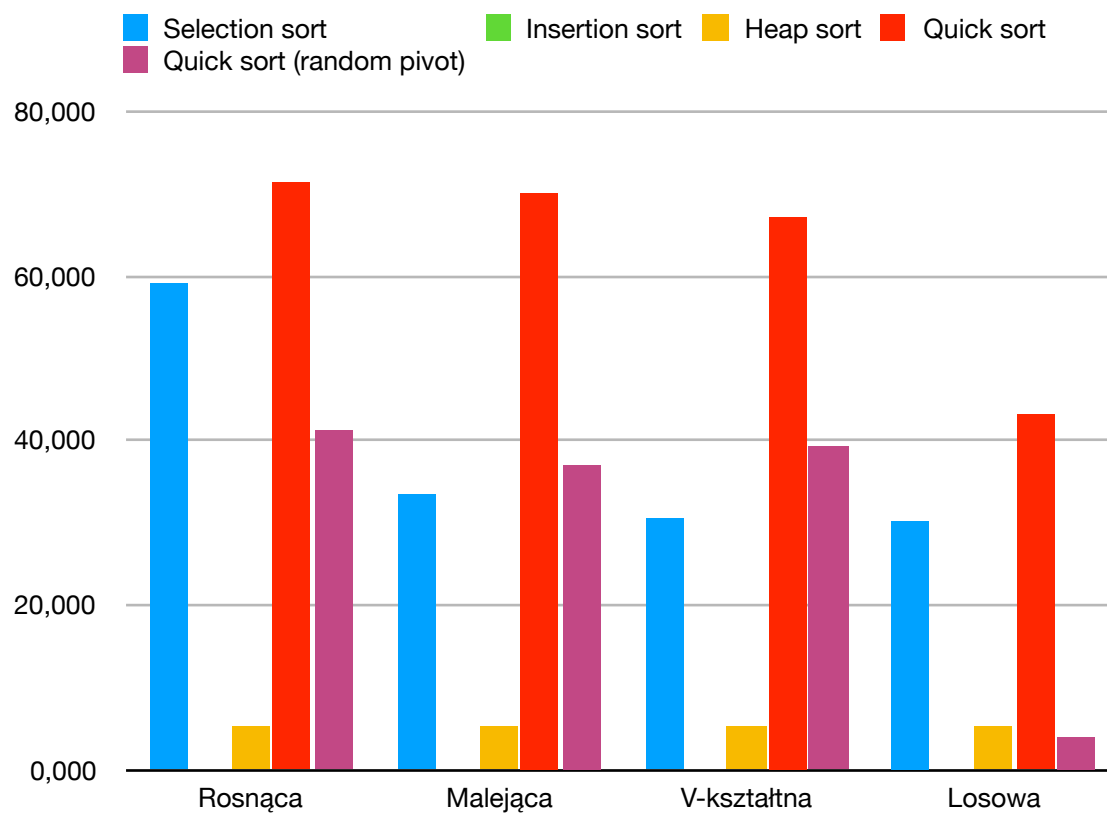
## 750 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	43.742 ms	0.142 ms	4.511 ms	47.713 ms	25.668 ms
<b>Tablica malejąca</b>	36.793 ms	0.169 ms	4.364 ms	47.559 ms	25.787 ms
<b>Tablica V-kształtna</b>	22.518 ms	0.141 ms	4.497 ms	48.387 ms	25.846 ms
<b>Tablica losowa</b>	21.381 ms	0.143 ms	4.36 ms	30.583 ms	3.217 ms



## 900 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	59.118 ms	0.171 ms	5.355 ms	71.273 ms	41.309 ms
<b>Tablica malejąca</b>	33.286 ms	0.169 ms	5.341 ms	69.948 ms	37.02 ms
<b>Tablica V-kształtna</b>	30.473 ms	0.168 ms	5.364 ms	67.157 ms	39.158 ms
<b>Tablica losowa</b>	30.085 ms	0.187 ms	5.335 ms	43.252 ms	3.804 ms



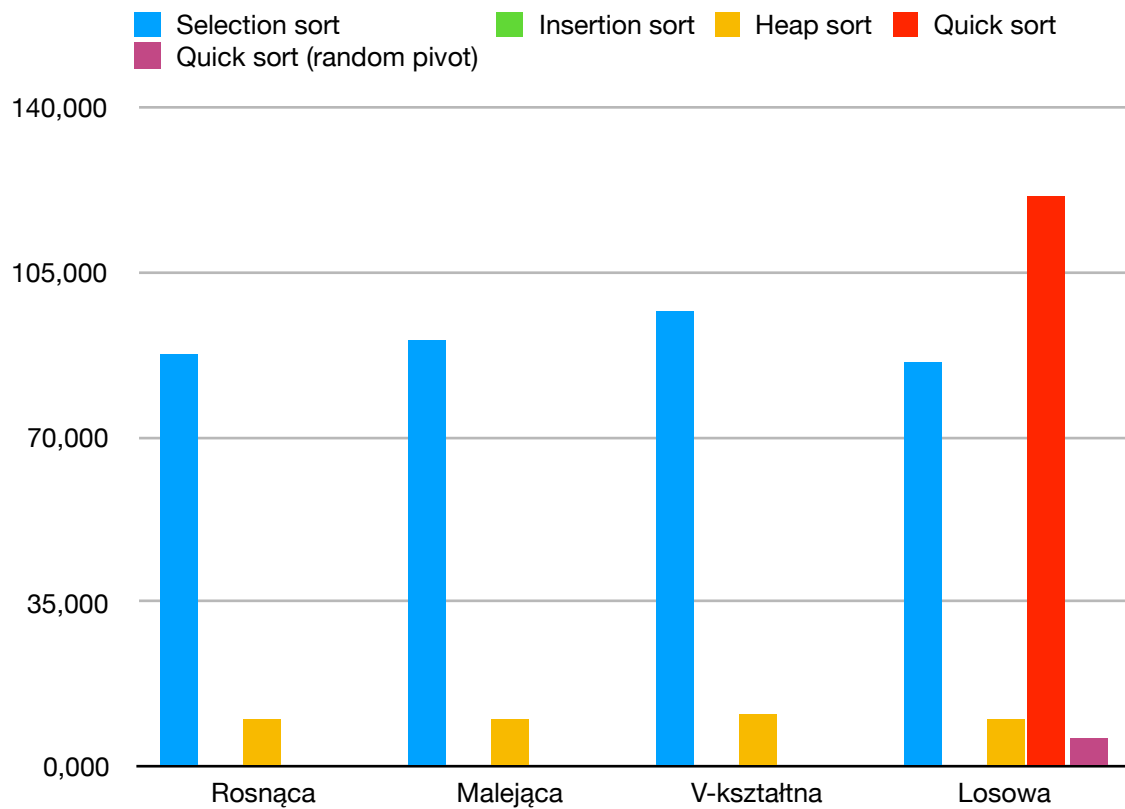


1500 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	87.576 ms	0.289 ms	9.867 ms	N/A*	N/A**
<b>Tablica malejąca</b>	90.425 ms	0.29 ms	9.983 ms	N/A*	N/A**
<b>Tablica V-kształtna</b>	96.525 ms	0.371 ms	10.911 ms	N/A*	N/A**
<b>Tablica losowa</b>	86.063 ms	0.295 ms	9.859 ms	120.85 ms	6,188 ms

N/A\* - Error in maximum recursion depth exceeded in comparison

N/A\*\* - Error in maximum recursion depth exceeded while calling a Python object

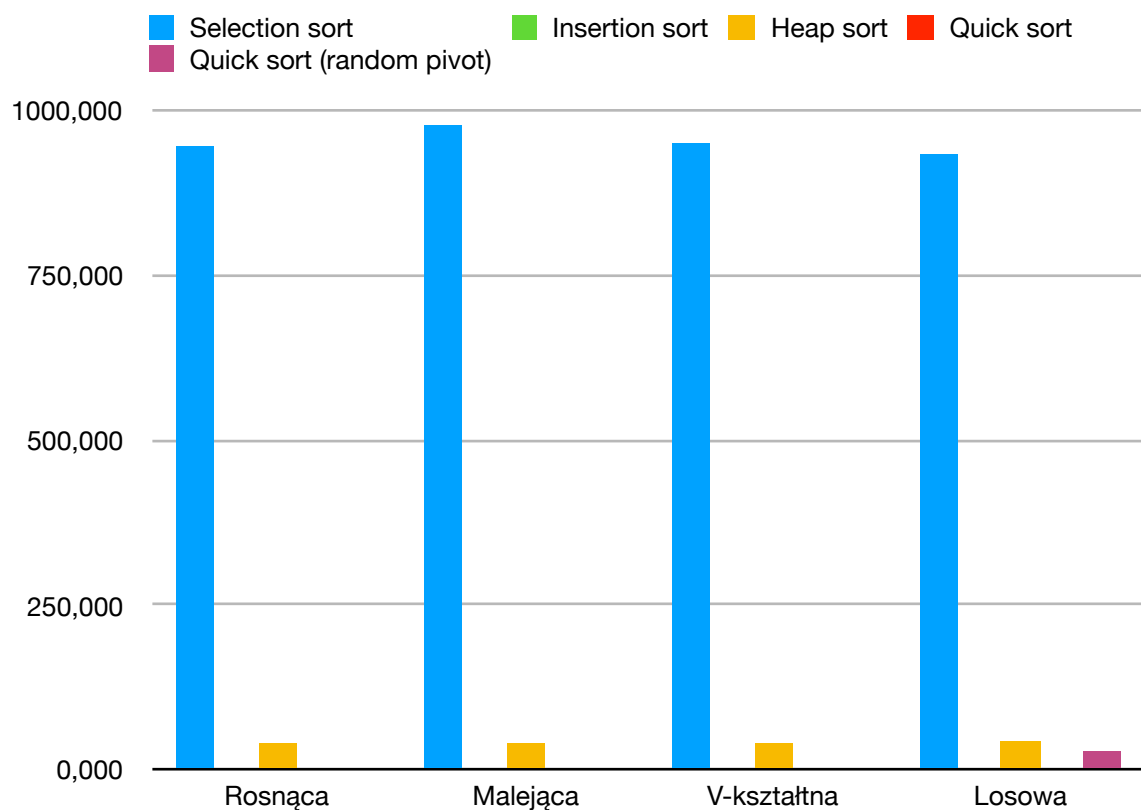


## 5000 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	945.765 ms	0.994 ms	38.706 ms	N/A*	N/A**
<b>Tablica malejąca</b>	977.607 ms	0.991 ms	38.823 ms	N/A*	N/A**
<b>Tablica V-kształtna</b>	951.539 ms	0.99 ms	40.07 ms	N/A*	N/A**
<b>Tablica losowa</b>	936.406 ms	1.015 ms	41.988 ms	N/A*	24.329 ms

N/A\* - Error in maximum recursion depth exceeded in comparison

N/A\*\* - Error in maximum recursion depth exceeded while calling a Python object

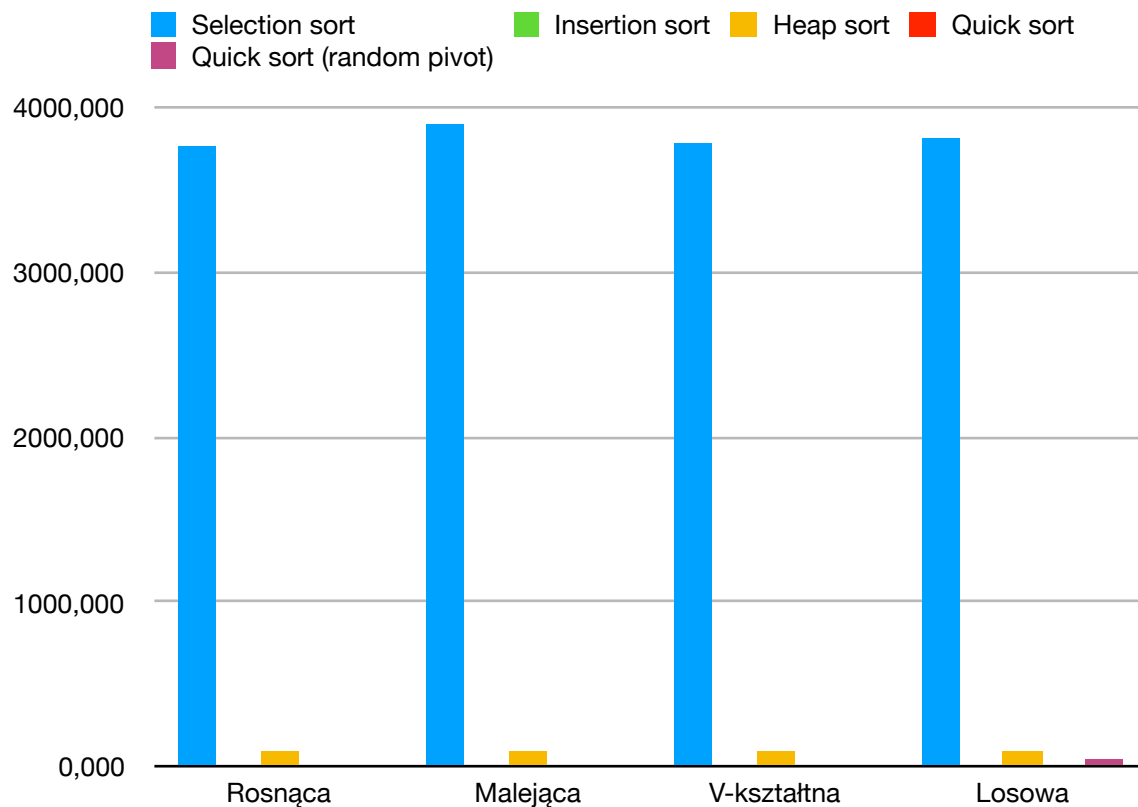


## 10000 elementowa tablica

	Selection sort	Insertion sort	Heap sort	Quick sort	Quick sort (random pivot)
<b>Tablica rosnąca</b>	3758.994 ms	1.973 ms	83.613 ms	N/A*	N/A**
<b>Tablica malejąca</b>	3889.975 ms	1.972 ms	85.12 ms	N/A*	N/A**
<b>Tablica V-kształtna</b>	3790.361 ms	1.968 ms	89.315 ms	N/A*	N/A**
<b>Tablica losowa</b>	3818.09 ms	2.113 ms	87.938 ms	N/A*	48.139 ms

N/A\* - Error in maximum recursion depth exceeded in comparison

N/A\*\* - Error in maximum recursion depth exceeded while calling a Python object



## Podsumowanie:

Najlepszy okazał się insertion sort. Dość zaskakująco zachowywał się quick sort który domyślnie traktowany jest jako jeden z najlepszych algorytmów. Heap sort zawsze zdawał się być w środku stawki. Wracając do quick sorta to zdecydowanie lepszym rozwiązaniem jest losowe położenie pivot'u. Jeżeli stanąłbym przed koniecznością użycia jakiegoś z algorytmów w prowadzonym projekcie to nie skorzystałbym z żadnego. Dlaczego? Przede wszystkim napisanie samemu dobrego algorytmu sortującego zajmuje czas a jego można by było poświęcić na inne czynności w projekcie. Dodatkowo na internecie znajduje się już pełno bibliotek z dobrze zaimplementowanymi algorytmami sortującymi np Numpy o niskiej złożoności obliczeniowej.