

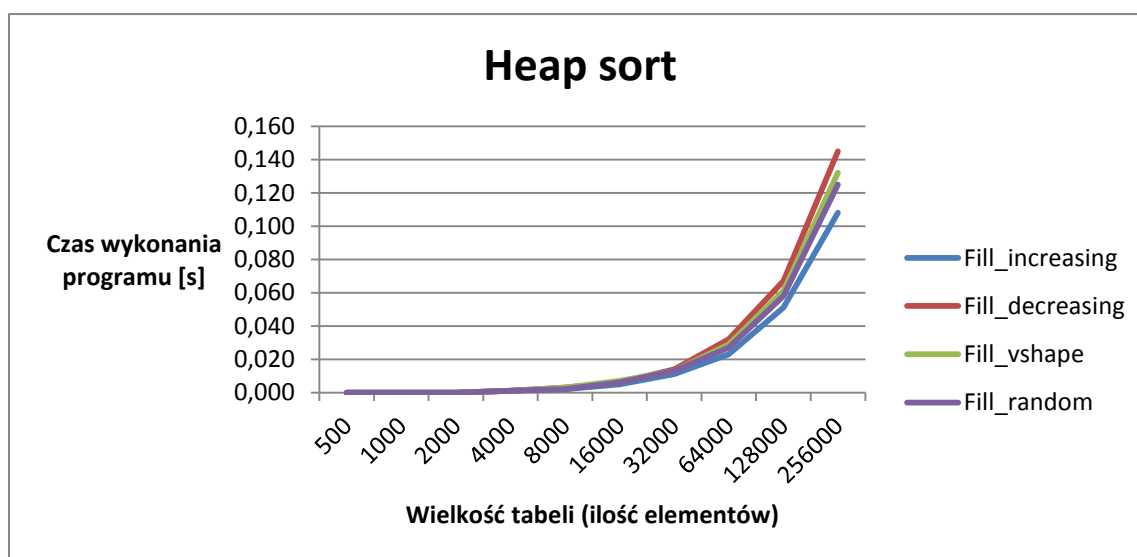
Sprawozdanie z algorytmów sortujących

Aleksander Pawlikowski

Antonina Stobińska

Heap sort

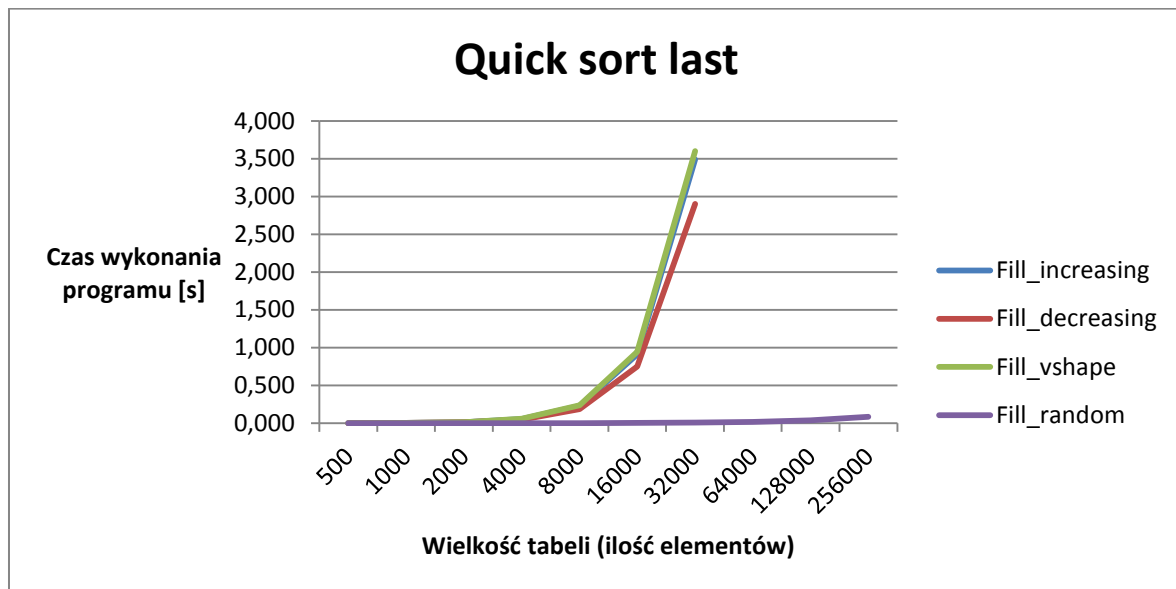
	500	1000	2000	4000	8000	16000	32000	64000	128000	256000
Fill_increasing	0	0	0	0,001	0,002	0,005	0,011	0,023	0,051	0,108
Fill_decreasing	0	0	0	0,001	0,003	0,006	0,014	0,032	0,067	0,145
Fill_vshape	0	0	0	0,001	0,003	0,007	0,013	0,029	0,061	0,132
Fill_random	0	0	0	0,001	0,002	0,006	0,013	0,027	0,058	0,125



- Algorytm wykonuje się w podobnym czasie dla wszystkich rozkładów danych, nie ważne czy są uporządkowane rosnąco czy malejąco, w najgorszym wypadku czas wykonania algorytmu jest nieznacznie dłuższy
- Z wykresu możemy odczytać złożoność logarytmiczną

Quick sort last (ostatnia wartość w tablicy jest elementem rozdzielającym)

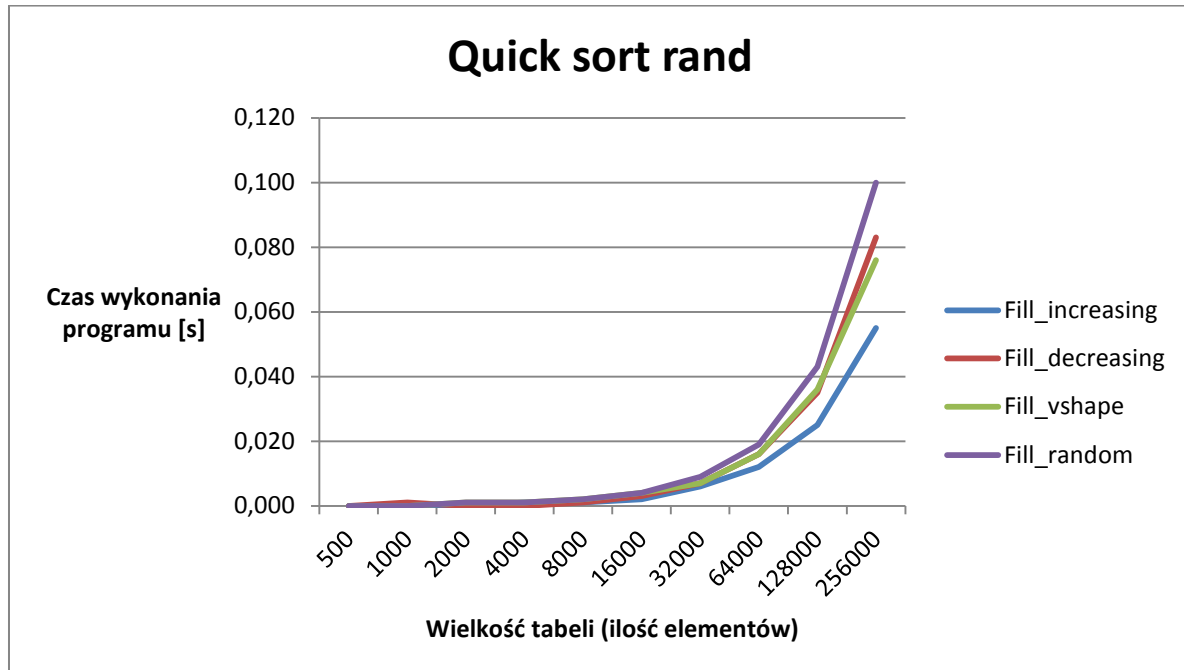
	<u>500</u>	<u>1000</u>	<u>2000</u>	<u>4000</u>	<u>8000</u>	<u>16000</u>	<u>32000</u>	<u>64000</u>	<u>128000</u>	<u>256000</u>
Fill_increasing	<u>0,001</u>	<u>0,004</u>	<u>0,014</u>	<u>0,057</u>	<u>0,227</u>	<u>0,911</u>	<u>3,491</u>			
Fill_decreasing	<u>0,001</u>	<u>0,003</u>	<u>0,011</u>	<u>0,047</u>	<u>0,187</u>	<u>0,75</u>	<u>2,902</u>			
Fill_vshape	<u>0,001</u>	<u>0,003</u>	<u>0,014</u>	<u>0,059</u>	<u>0,236</u>	<u>0,943</u>	<u>3,599</u>			
Fill_random	<u>0</u>	<u>0,001</u>	<u>0</u>	<u>0,001</u>	<u>0,002</u>	<u>0,004</u>	<u>0,008</u>	<u>0,018</u>	<u>0,038</u>	<u>0,085</u>



- Algorytm wykonuje się w podobnym czasie dla prawie wszystkich rozkładów danych wejściowych, nie ważne czy są uporządkowane rosnąco czy malejąco.
- W najgorszym wypadku czas wykonania algorytmu jest nieznacznie dłuższy. Jedynie dla losowych danych jest najszybciej wykonany.
- Z wykresu możemy odczytać złożoność kwadratową dla prawie wszystkich rozkładów danych.

Quick sort rand (losowa wartość z tablicy jest elementem rozdzielającym)

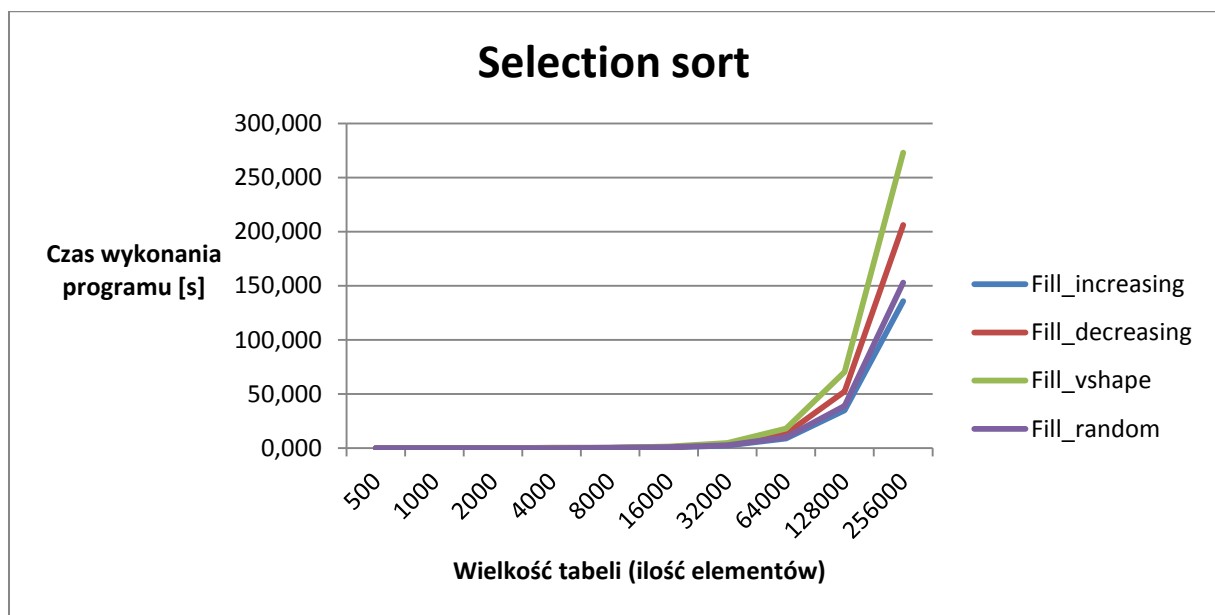
	500	1000	2000	4000	8000	16000	32000	64000	128000	256000
Fill_increasing	0	0	0	0,001	0,001	0,002	0,006	0,012	0,025	0,055
Fill_decreasing	0	0,001	0	0	0,001	0,003	0,007	0,016	0,035	0,083
Fill_vshape	0	0	0,001	0,001	0,002	0,004	0,007	0,016	0,036	0,076
Fill_random	0	0	0,001	0,001	0,002	0,004	0,009	0,019	0,043	0,1



- Algorytm wykonuje się w podobnym czasie dla wszystkich rozkładów danych wejściowych, nie ważne czy są uporządkowane rosnąco czy malejąco, w najgorszym wypadku czas wykonania algorytmu jest nieznacznie dłuższy, są jednak małe różnice w czasie
- Z wykresu możemy odczytać złożoność logarytmiczną

Selection sort

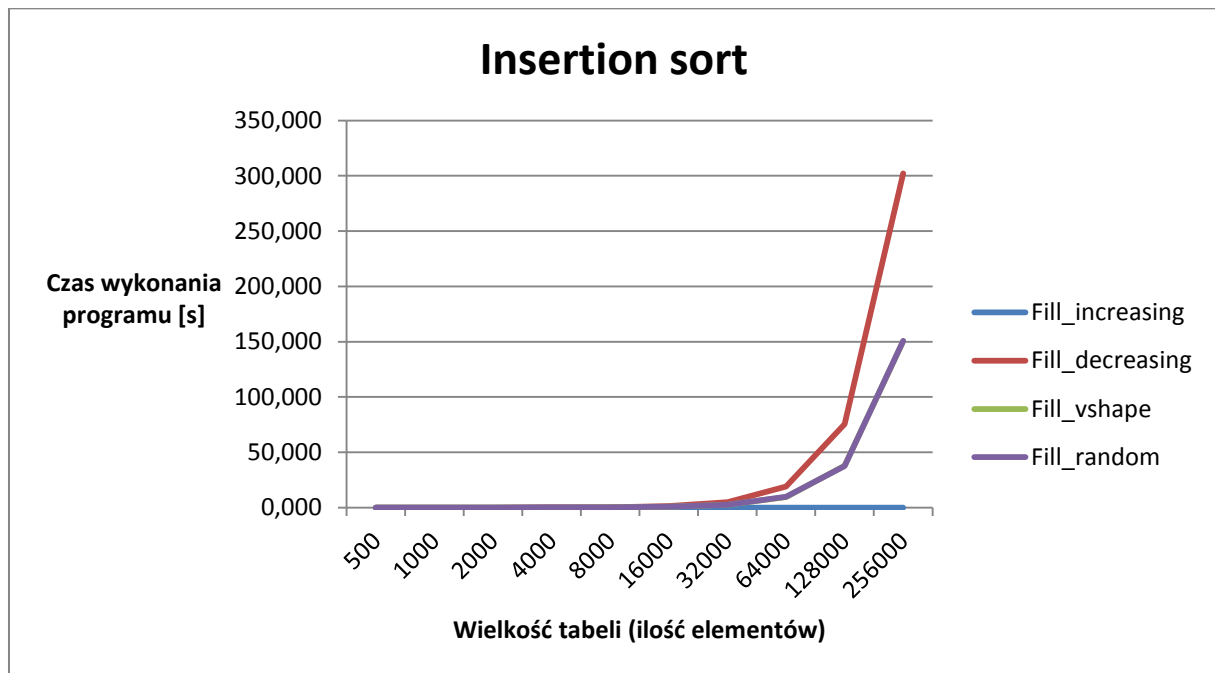
	500	1000	2000	4000	8000	16000	32000	64000	128000	256000
Fill_increasing	0	0	0,015	0,031	0,157	0,593	2,313	8,703	34,686	135,651
Fill_decreasing	0	0	0,015	0,063	0,234	0,906	3,453	13,234	52,217	206,117
Fill_vshape	0	0,015	0,016	0,078	0,312	1,235	4,656	17,859	70,2	273,098
Fill_random	0	0	0,016	0,031	0,187	0,672	2,672	10,031	39,014	152,978



- Algorytm wykonuje się w podobnym czasie dla wszystkich rozkładów danych, nie ważne czy są uporządkowane rosnąco czy malejąco.
- W najgorszym wypadku (czyli w posortowanym w sposób v shape rozkładzie danych) czas wykonania algorytmu jest nieznacznie dłuższy; czas wykonywania algorytmu szybko rośnie dla większej ilości danych
- Z wykresu możemy odczytać złożoność kwadratową w najgorszym wypadku

Insertion sort

	500	1000	2000	4000	8000	16000	32000	64000	128000	256000
Fill_increasing	0	0	0	0	0	0	0	0	0	0
Fill_decreasing	0	0,016	0,015	0,078	0,328	1,313	4,89	19	75,34	302,035
Fill_vshape	0	0	0,016	0,031	0,172	0,641	2,562	9,593	37,827	150,385
Fill_random	0	0,015	0	0,047	0,156	0,641	2,562	9,609	37,639	150,681



- W tym algorytmie widać duże rozbieżności czasowe między różnymi rozkładami danych
- Dla uporządkowanych rosnąco nie zajmuje on czasu
- W najgorszym wypadku, czyli dane uporządkowane malejąco, algorytm wykonuje się bardzo długo i szybko rośnie jego czas wykonywania zależnie od ilości danych, w tym wypadku jest to złożoność kwadratowa