

Liste med helt tilfeldige verdier:

	Quicksort	Quicksort /m innsetningssort	Dual pivot quicksort	Dual pivot quicksort /m innsetningssort
n = 1 000 00	15625 μ s	15625 μ s	15625 μ s	15625 μ s
n = 1 000 000	109 ms	109 ms	117 ms	109 ms
n = 10 000 000	1.375 s	1.296 s	1.312 s	1.250 s
n = 100 000 000	15.796 s	14.671 s	15.078 s	14.406 s

Liste med mange duplikater:

	Quicksort	Quicksort /m innsetningssort	Dual pivot quicksort	Dual pivot quicksort /m innsetningssort
n = 1 000 00	15625 μ s	15625 μ s	15625 μ s	15625 μ s
n = 1 000 000	78 ms	62 ms	76 ms	68 ms
n = 10 000 000	796 ms	687 ms	812 ms	703 ms
n = 100 000 000	8.609 s	7.703 s	8.921 s	7.828 s

Sortert liste:

	Quicksort	Quicksort /m innsetningssort	Dual pivot quicksort	Dual pivot quicksort /m innsetningssort
n = 1 000 00	15625 μ s	15625 μ s	93 ms	62 ms
n = 1 000 000	46875 μ s	31250 μ s	1.359 s	1.343 s
n = 10 000 000	484 ms	406 ms	16.703 s	16.578 s
n = 100 000 000	5.562 s	4.687 s		

Konklusjon:

Ved lister med helt tilfeldig data, ser man at dual pivot quicksort med innsetningsort var raskest når n var stor. Brukte innsetningsort når listen var splittet opp til en lengde som var mindre eller like 50(testet litt grovt).

Ved lister med mange duplikater var quicksort med innsetningsort den med best tid på høy og lavere n. Brukte innsetningsort når listen var splittet opp til en lengde som var mindre eller like 40(testet litt grovt).

Fikk ikke endret dual pivot quicksort algoritmen til å bruke to pivots mer sentralisert i listen. Derfor under sorterte lister, ble dual pivot tidtakingen mye større av grunnen forklart i oppgaven. Raskest var Quicksort med innsetningssort.