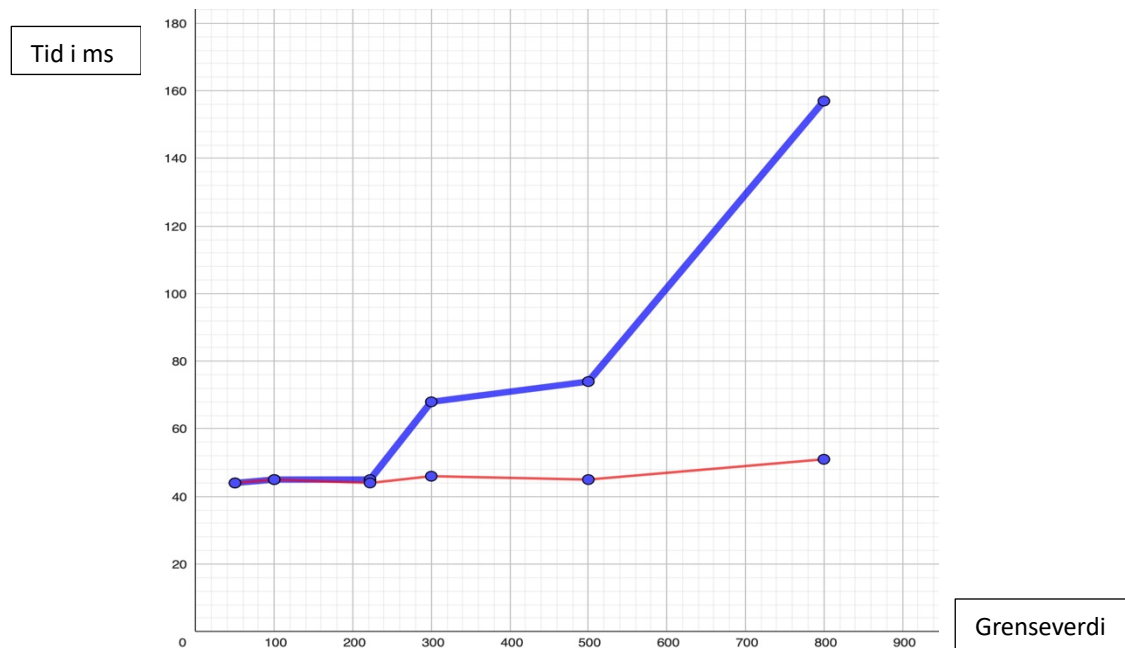


Øving 3 Algdad

Oppgave 1

Måleserie



Grafen illustrer tidskompleksiteten til de ulike variantene av quicksort-algoritmene når grenseverdien endres. Det er viktig å merke seg at det bare er grenseverdien til quicksort-algoritmen med innsettingssortering (blå graf) som endres, og dette fører til en økning i tidsforbruket. Standardutgaven av quicksort-algoritmen (rød graf) derimot, holder en konstant grenseverdi, men den er inkludert i grafen for å hjelpe med å finne en passende størrelse for overgangen til hjelpealgoritmen i den andre quicksort-algoritmen.

Quicksort-algoritmen med innsettingssortering starter litt raskere enn standardutgaven (se tidtaking for tydeligere målinger), men opplever deretter en drastisk økning i tidsforbruk når grenseverdien økes. Dette skyldes at innsettingssortering har en kjøretid på $O(n^2)$, og når quicksort-algoritmen med gjennomsnittlig kjøretid på $O(n \log n)$ bytter til innsettingssortering tidligere, vil det påvirke kjøretiden betydelig. Vi kan se at kjøretiden til quicksort-algoritmen med innsettingssortering blir høyere enn standardutgaven når grenseverdien passerer 222.

Tidtaking

```
Quicksort with helper method.
Elements: 1000000
Threshold: 50
Time: 44 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 46 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method.
Elements: 1000000
Threshold: 100
Time: 45 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 50 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method.
Elements: 1000000
Threshold: 222
Time: 45 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 44 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method.
Elements: 1000000
Threshold: 300
Time: 68 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 46 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method.
Elements: 1000000
Threshold: 500
Time: 74 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 45 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method.
Elements: 1000000
Threshold: 800
Time: 157 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 51 ms.
Sorted: true
Equal sum before and after sorting: true
```

Samme trenden blir demonstrert her som i grafen. Quicksort med innsettingssortering starter med lavere tidsforbruk sammenlignet med standardutgaven, og deretter øker den drastisk når grenseverdien øker. I tillegg er det presentert to ekstra metoder for å bekrefte om listen faktisk er sortert, og summen forblir uendret før og etter sorteringen.

Sortere sortert tabell

Bildet illustrerer hvordan algoritmene håndterer lister av data, både når listene ikke er sortert og når de er sortert. Som vi kan observere, er kjøretiden hos både quicksort-algoritmen med innsettingssortering og uten lavere når listen allerede er stotret.

```
Quicksort with helper method.
Elements: 1000000
Threshold: 40
Time: 44 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method.
Elements: 1000000
Time: 46 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort with helper method on sorted array.
Elements: 1000000
Threshold: 40
Time: 24 ms.
Sorted: true
Equal sum before and after sorting: true
```

```
Quicksort without helper method on sorted array.
Elements: 1000000
Time: 23 ms.
Sorted: true
Equal sum before and after sorting: true
```