# Sorteringsexperimentet

## Utvärdering av uppgift 2a)

**Resultat**

Det utfördes en binär sökning på M (element som skickas till Insertionsort), varje M testades 10 gånger och det togs fram ett snittvärde för att kunna se mönster och vilket M som är det bästa. När det kom till Quicksort hade den minst körtid när M = 16 - 32. Det upptäcktes ingen stor skillnad på att använda shuffle eller inte, resultatet visar att det blev bättre snittvärde att ha ingen shuffle men skillnaden var så liten och resultatet ändras för varje gång det testades även om man det var ett snittvärde på tio värden för varje test. Mergesort hade den minsta körtiden när M = 16.

**Diskussion**

Anledningen till att det går snabbare att byta till Insertionsort för små tal men blir långsammare för stora tal är för att den bästa tidskomplexiteten för insertionsort är O(n) medan quicksort och mergesort bästa tidskomplexitet är O(n log n) vilket är långsammare. Som medel tidskomplexitet har insertionsort O(n^2) och quicksort och mergesort har medel tidskomplexitet O(n log n) vilket är snabbare, vilket innebär att desto mer insertionsort behöver jobba desto mindre effektiv är den. Om arrayen är mer sorterad när bytet sker kommer insertionsort att få en bättre körtid.

## Quicksort (Snittvärde av 10 tester)

| M | N = data.Length Med Shuffle | N = data.Length Utan Shuffle | N = data.Length/2 Med Shuffle | N = data.Length/2 Utan Shuffle |
|---|---|---|---|---|
| 2 | 0,42 | 0,332 | 0,225 | 0,177 |
| 4 | 0,389 | 0,306 | 0,172 | 0,149 |
| 8 | 0,386 | 0,265 | 0,150 | 0,138 |
| 16 | 0,31 | 0,254 | 0,144 | 0,134 |
| 32 | 0,363 | 0,252 | 0,147 | 0,133 |
| 64 | 0,409 | 0,277 | 0,158 | 0,141 |
| 128 | 0,544 | 0,344 | 0,194 | 0,173 |
| 256 | 0,653 | 0,495 | 0,296 | 0,256 |
| 512 | 1,021 | 0,873 | 0,496 | 0,467 |

# Mergesort (Snittvärde av 10 tester)

| M | N = data.length() | N = (data.length()/2) |
|---|---|---|
| 0 | 0,6 | 0,36 |
| 2 | 0,52 | 0,25 |
| 4 | 0,48 | 0,24 |
| 8 | 0,46 | 0,24 |
| 16 | 0,45 | 0,22 |
| 32 | 0,47 | 0,25 |
| 64 | 0,58 | 0,28 |
| 128 | 0,76 | 0,375 |
| 256 | 1,25 | 0,60 |