

CPU: 1.4 GHz Intel Core i5 (4 kjerner)

Tid oppgitt i ms

n	Array.sort()	Sekvensiell	Parallell
1k	0.617044	0.5029	0.828658
10k	4.902349	1.05784	0.858346
100k	42.67261	3.955199	0.835645
1m	158.292875	8.0022	1.057829
10m	1513.696491	22.588298	4.19055

I min besvarelse har jeg ikke med en graf, skjønte ikke helt hvordan jeg skulle lage den, skal spørre på neste gruppe time. Jeg har ikke lagd forskjellige filer for de forskjellige sorteringene, har heller lagd metoder. På $n = 100m$ fikk jeg "out of memory", har java8 64-bit, kjørte med all RAM jeg hadde og fikk allikevel error. Så $n = 10m$ var det beste jeg klarte.

Med disse resultatene ser vi aldri at den parallelle koden gir speedup > 1 , men kjøringstiden øker mer og mer på `Array.sort()` og den Sekvensielle koden, mens den parallelle delen ikke øker like drastisk. Vil regne med at rundt $100m - 1000m$ vil den parallelle koden gi speedup > 1 . fikk dessverre ikke testet dette.

Testet litt rundt for å se om jeg klarte å speedup'e den parallelle delene. Fant ut at det som tar lengst tid er `CyclicBarrier` som tar lengst tid, fjernet denne og fikk den parallelle delen til å ta rundt 1-2ms, men da ble det mye feil i array'et. Men vil si at feilene var ikke så stor.

Re: Fikset Kjøringstiden og oppdaterte tabellen, lagret alle tallene øverst i indekseringen til vær tråd, main tråden samler alt helt i toppen til slutt.