

Denne rapporten er om min innlevering av Oblig3 i INF2440 for både sekvensiell og parallell løsning av oppgaven.

Oppgaven brukes som slik:

JavaC *.java

Java Oblig2 n k

N er hvilket tall du skal finne primtall under

K er antall tråder du vil bruke hvis k er 0 settes alle tilgjengelige tråder i arbeid.

Ved implementering av den sekvensielle biten av leting etter primtall og primtallsfaktorisering tok jeg utgangspunkt i prekoden som ble gitt, men syntes det var lettere å bruke et boolean array i stedet for et byte array samtidig som det gjorde den sekvensielle delen min betraktelig raskere så jeg la bort den delen og lagde en oppgave nesten fra scratch, og hentet kun inspirasjon fra prekoden. Noe jeg ikke tenkte på når jeg gjorde dette byttet var jo at boolean tar opp mye mer plass i minnet enn en byte array, så ikke noe krav i oppgaven om lagringsmåte, men det er jo rimelig klart for meg i ettertid at det var en dårlig løsning jeg har valgt ettersom n blir høyere så gir det problemer med minnebruk og jeg ville nok laget en løsning som brukte mindre plass per tall hvis jeg skulle startet på nytt nå. En annen måte kunne vært å dele inn primtallene mindre enn roten av n til hver tråd og de så kun markerte av de tråden har fått tildelt i hele arrayet, i stedet for å kun se på sitt intervall med alle primtallene, dette burde spart tid ettersom det hadde blitt mindre løkker, burde nok også laget en egen monitor eller noe slikt i stedet for å bruke cyclic barrier kunne spart tid under letingen etter primtall.

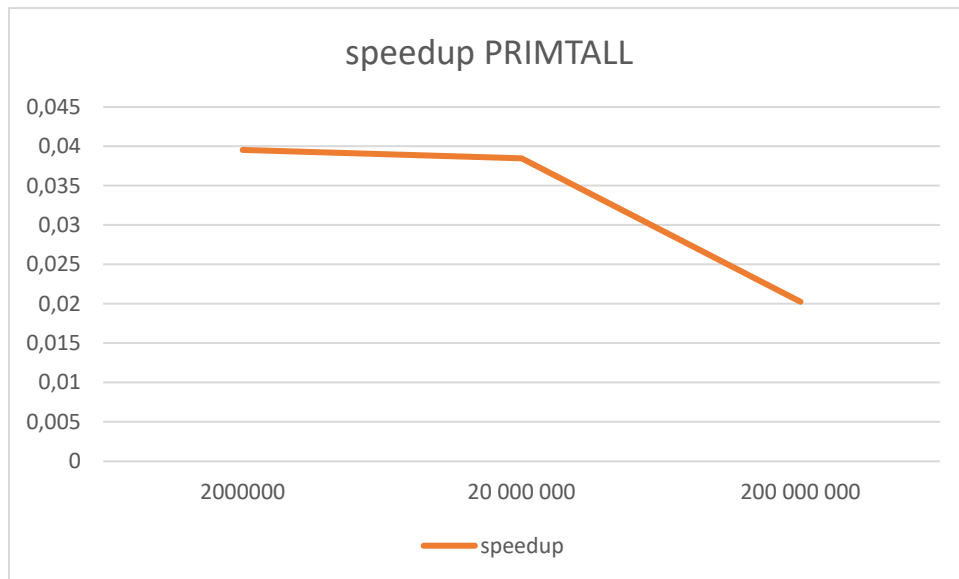
Jeg mangler også data for $N = 2 * 10^9$ siden jeg ikke hadde tid til å vente på at kjøringen skulle bli ferdig. Faktoriseringen min funker ikke har bare fått testet den når $n = 100$ og finner noen faktorer som ikke er primtall, men det er nok kun kalkulerings/slurvefeil ved sammenlegging i slutten av `initiateFactorization()` metoden. Bruk av `ArrayList<Long>` og `ArrayList<Long, ArrayList<Long>>` er heller ikke heldig for minne bruk og måtte allokere mer minne til heap space allerede på $2 * 10^6$.

Jeg har forståelse for at denne innleveringen ikke kommer til å bli godkjent, men håper det er vist nok til å få et forsøk til på oppgaven, all tilbakemelding tas i mort med glede, tidene for parallell er ikke av mer enn 1 kjøring, men jeg forventer ikke å se store utslag ved flere kjøringer, men den sekvensielle delen er kjørt 7 ganger slik det står nedenfor.

Tidsenhet = ms.

Mediantid av 7 kjøringer.

N	Sekvensiell		Parallell		speedup	
	Primtall	Faktorisering	Primtall	Faktorisering	Primtall	Faktorisering
2 000 000	6.455299	108.694755	163.246478	N/a	0,06	N/a
20 000 000	175.986299	792.567995	4021.105309	N/a	0,222	N/a
200 000 000	2379.616742	6950.832586	110193.790118	N/a	0,0215	N/a
2 000 000 000	28991.51819	35167.336563	N/a	N/a	N/a	N/a



N	Sekvensiell		Parallell		speedup	
	Primtall	Faktorisering	Primtall	Faktorisering	Primtall	Faktorisering
2 000 000	12.339252	368.905142				
20 000 000	462.154876	2375.312544				
200 000 000	3405.369513	4607.462682				
2 000 000 000	58017.51615	19980.863611				

Manuell sjekk av alle primtall under 100

Number of primes less than N found: 25 All primes less than N(100):		Sequence	Prime Number
01 02		1	2
02 03		2	3
03 05		3	5
04 07		4	7
05 11		5	11
06 13		6	13
07 17		7	17
08 19		8	19
09 23		9	23
10 29		10	29
11 31		11	31
12 37		12	37
13 41		13	41
14 43		14	43
15 47		15	47
16 53		16	53
17 59		17	59
18 61		18	61
19 67		19	67
20 71		20	71
21 73		21	73
22 79		22	79
23 83		23	83
24 89		24	89
25 97		25	97