

Grupp 20

Prestanda på datastrukturer; Linked List och TreeSet

André Frisk, Gustav Lindqvist, Erik Lundberg & Axel Tobieson Rova

Frågeställning

- Hur väl överensstämmer mätningarna med det teoretiska resultatet vid tilläggning?
- Hur väl överensstämmer mätningarna med det teoretiska resultatet vid sökning?
- Blir resultatet utan JIT-kompilator likt det teoretiska i jämförelse med utan JIT?
- Vilken datastruktur presterar bäst vid given indata?



Metod

- Testar insättning och sortering
- 3 indata filer med 1000, 10 000 respektive 100 000 osorterade heltal.
- 30 körningar av varje indata med JIT av respektive på.
- Summera och validera i R



Slutsats

- Tidskomplexiteter för operationer: $O(1)$, $O(\log n)$, $O(n)$
- Insättning för Linked List var snabbare
- Sökning för TreeSet var snabbare
- Utan JIT kompilator gav betydligt långsammare exekveringstid



10.000 - List Searching

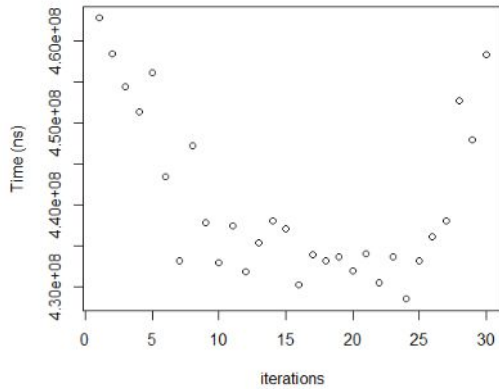


Fig. 3. Sökning i Linked List, 10 000 tal

10.000 - Tree Searching

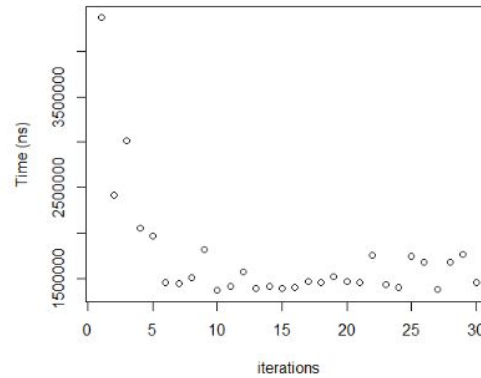


Fig. 5. Sökning i TreeSet, 10 000 tal

10.000 - List Adding

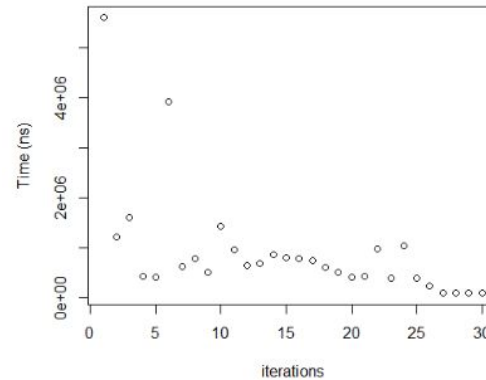


Fig. 2. Insättning i Linked List, 10 000 tal

10.000 - Tree Adding

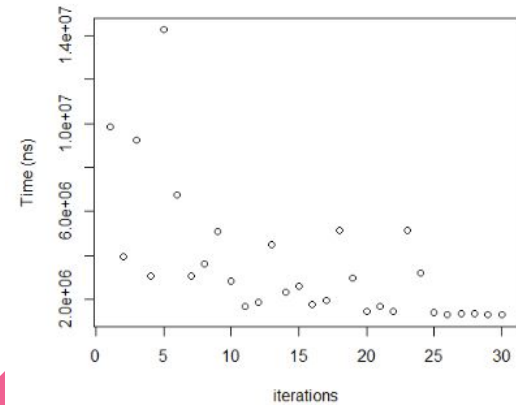
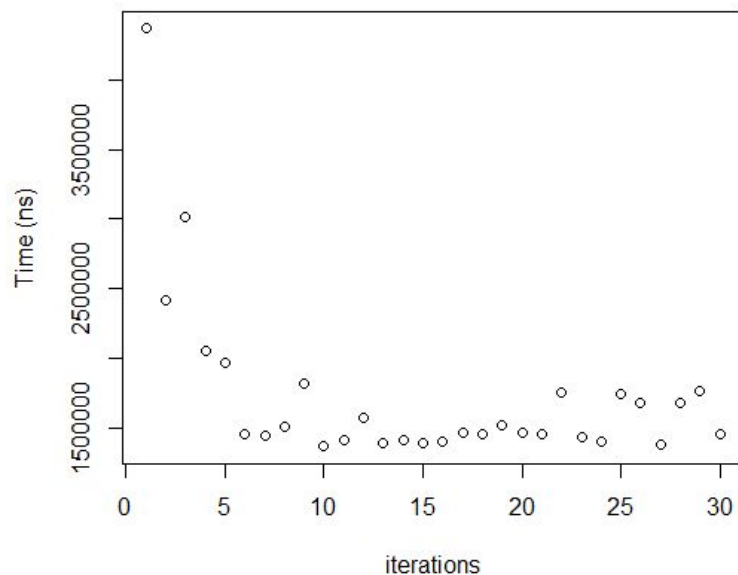


Fig. 4. Insättning i TreeSet, 10 000 tal

Validitetshot

10.000 - Tree Searching



Delar på N för att få ut tid för varje					
Mean tid/n	1000	10000	100000	1000 utan JIT	10000 utan JIT
List insättning	264.79	91.9507	18.40153	1019.4	982.6217
List sökning	3599.93	44040.2333	1.97E+05	69219.54	5.64E+05
Tree Insättning	1242.017	359.278	300.1117	4385.857	4760.9183
Tree sökning	349.16	172.4357	217.19563	3125.547	3757.7763
Median tid/n	1000	10000	100000	1000 utan JIT	10000 utan JIT
List insättning	125.05	64.26	10.8005	886.7	948.11
List sökning	2966.1	43653.235	1.24E+05	69146.5	5.60E+05
Tree Insättning	358.85	272.715	263.866	4095.15	4668.48
Tree sökning	194.1	146.405	212.6545	2897.2	3672.265