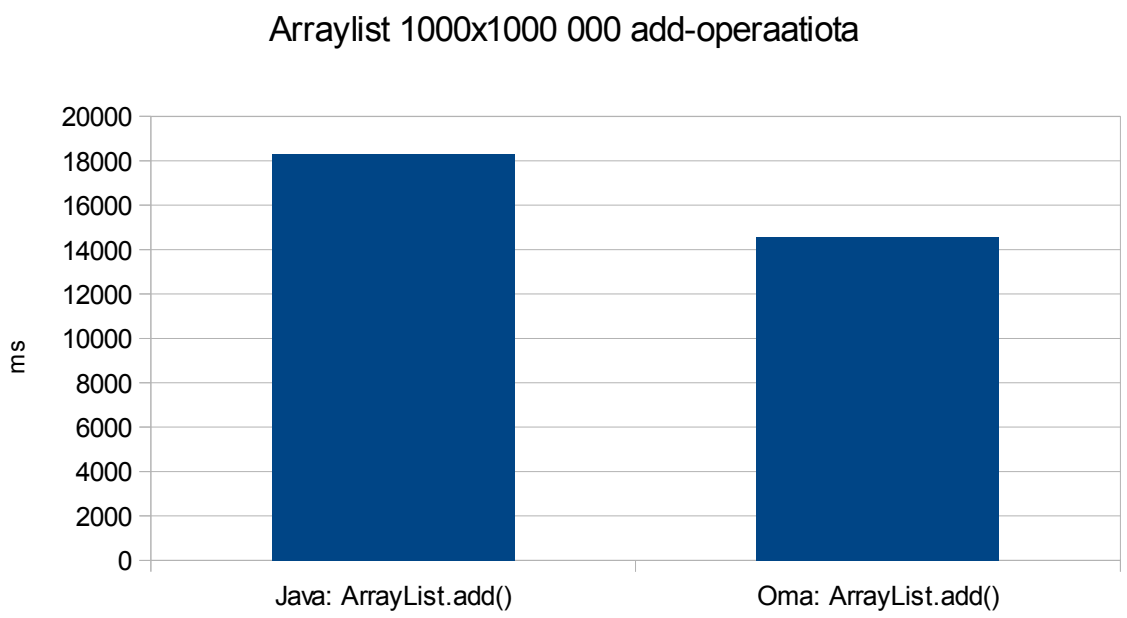
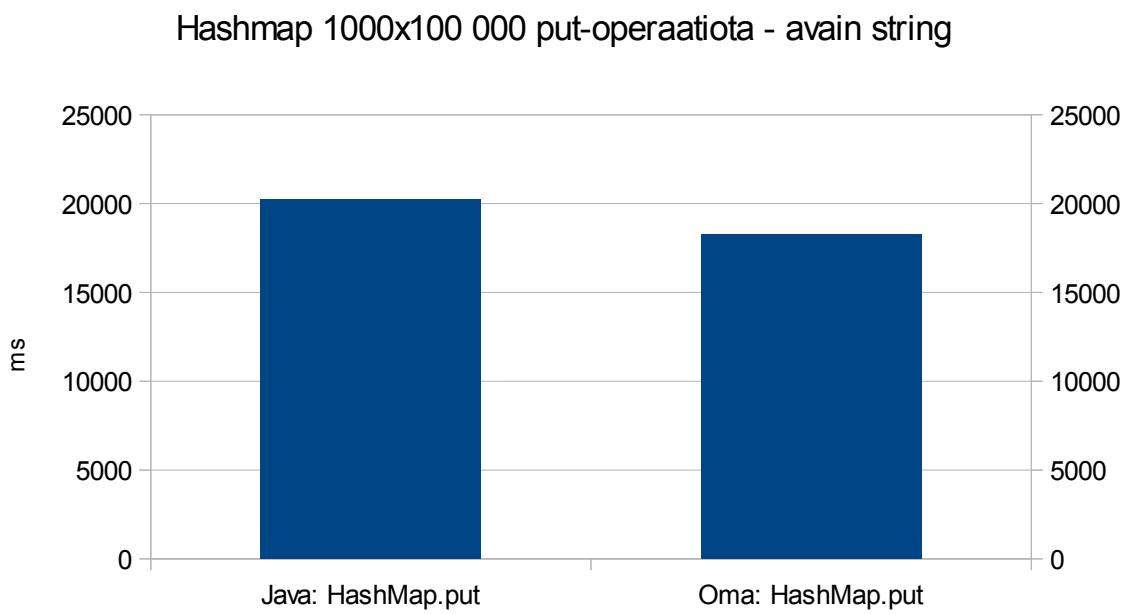
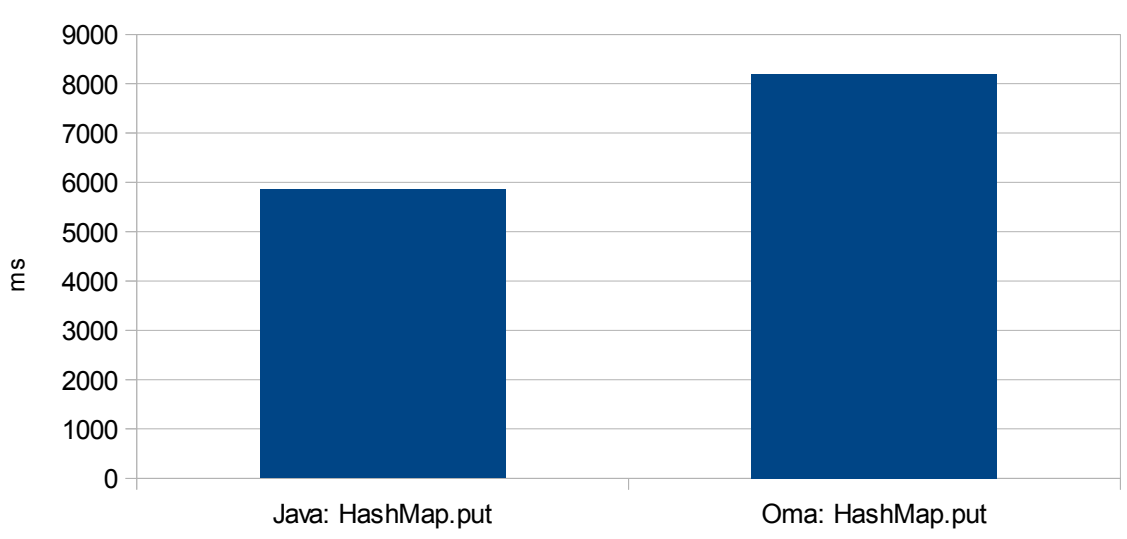
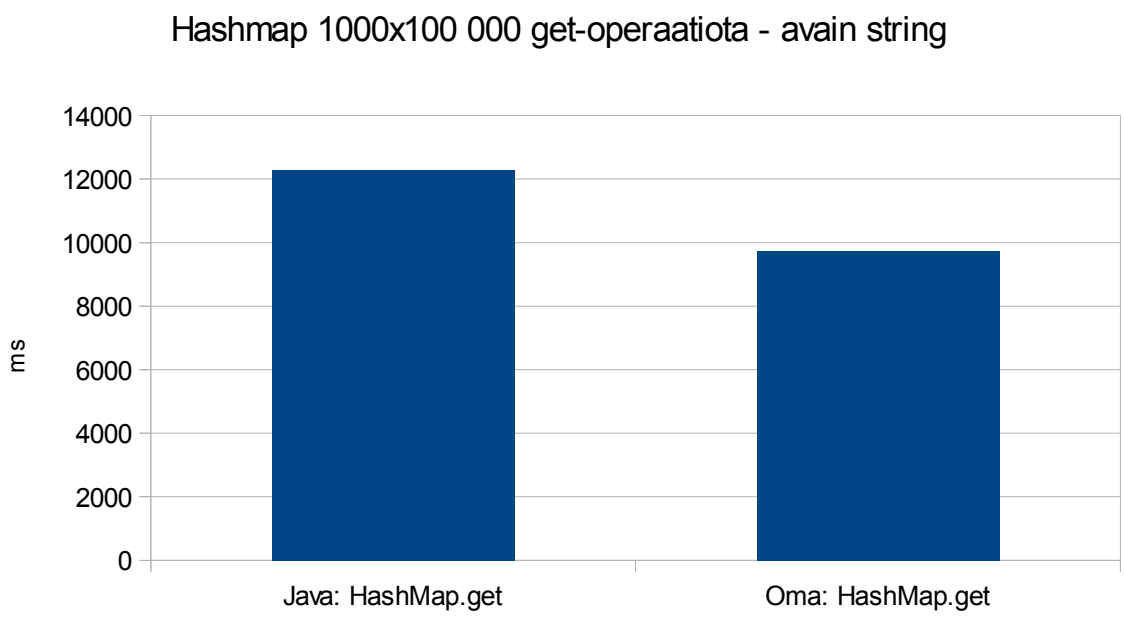
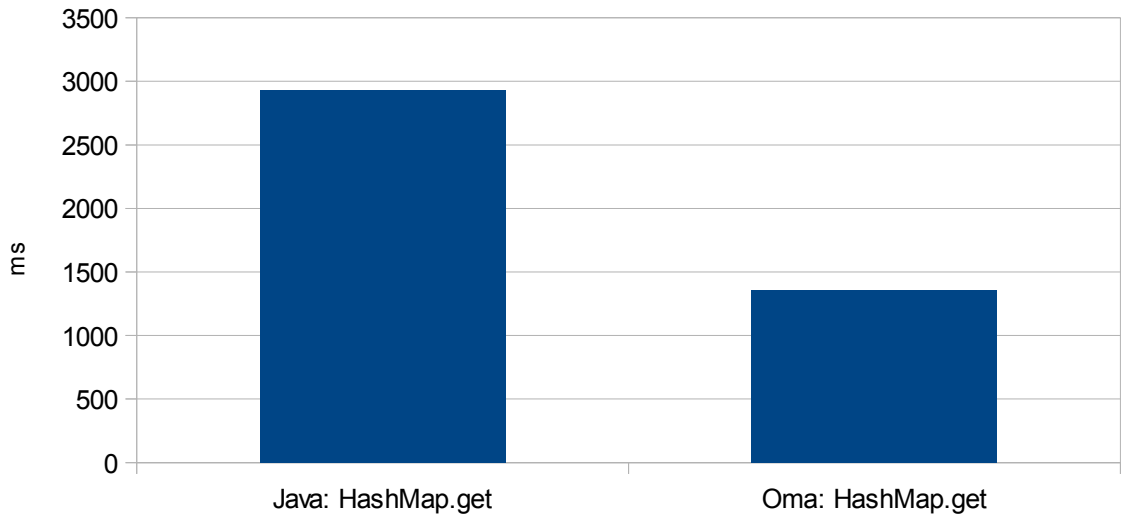
Javan tietorakenteet vs omat tietorakenteet

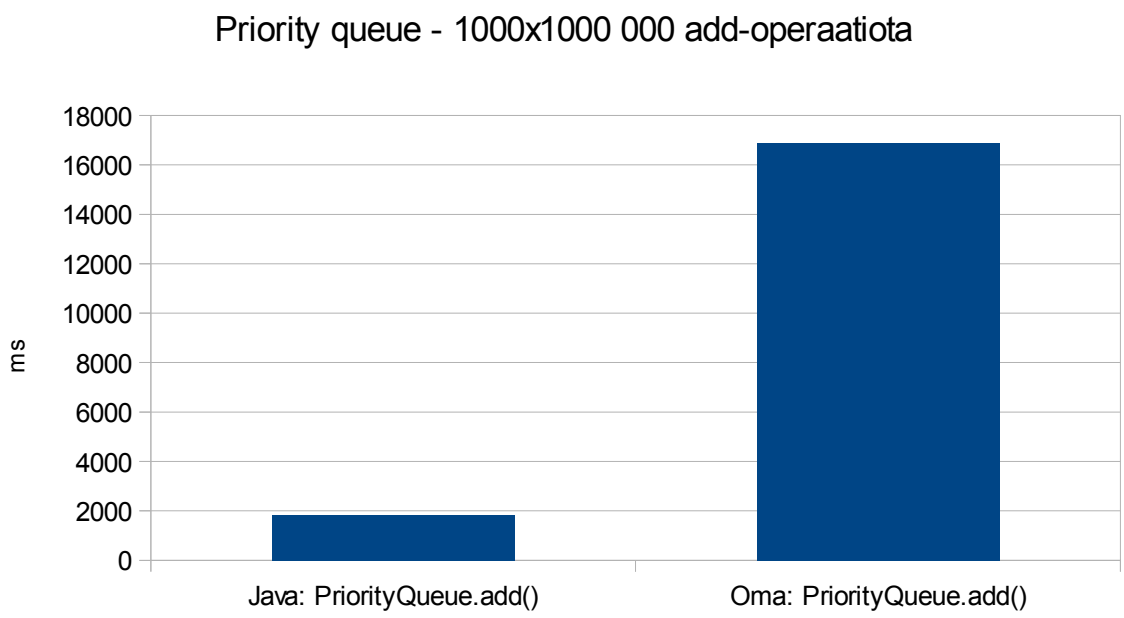


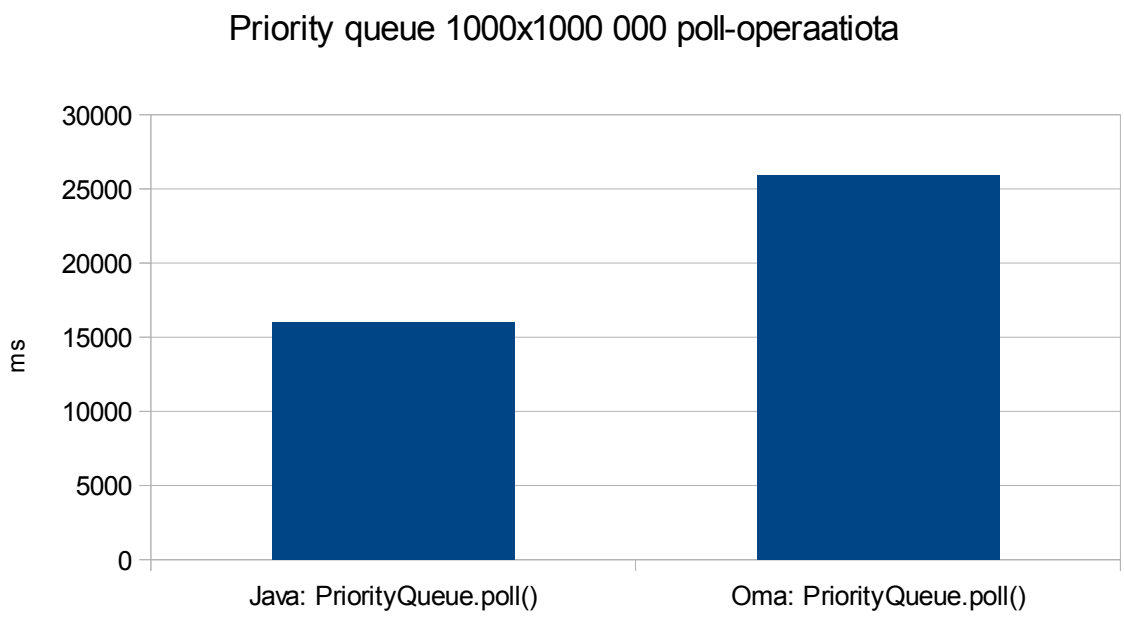





Hashmap 1000x100 000 get-operaatiota - avain string
ms
14000
12000
10000
8000
6000
4000
2000
0
Java: HashMap.get
Oma: HashMap.get

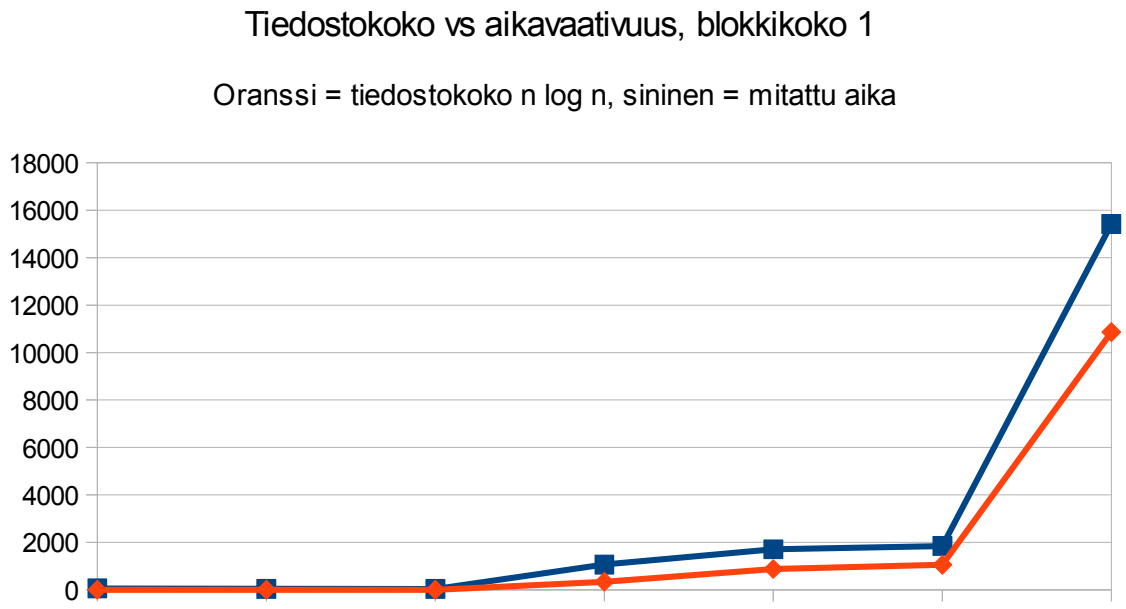

Hashmap 1000x100 000 get-operaatiota - avain Integer
ms
3500
3000
2500
2000
1500
1000
500
0
Java: HashMap.get
Oma: HashMap.get

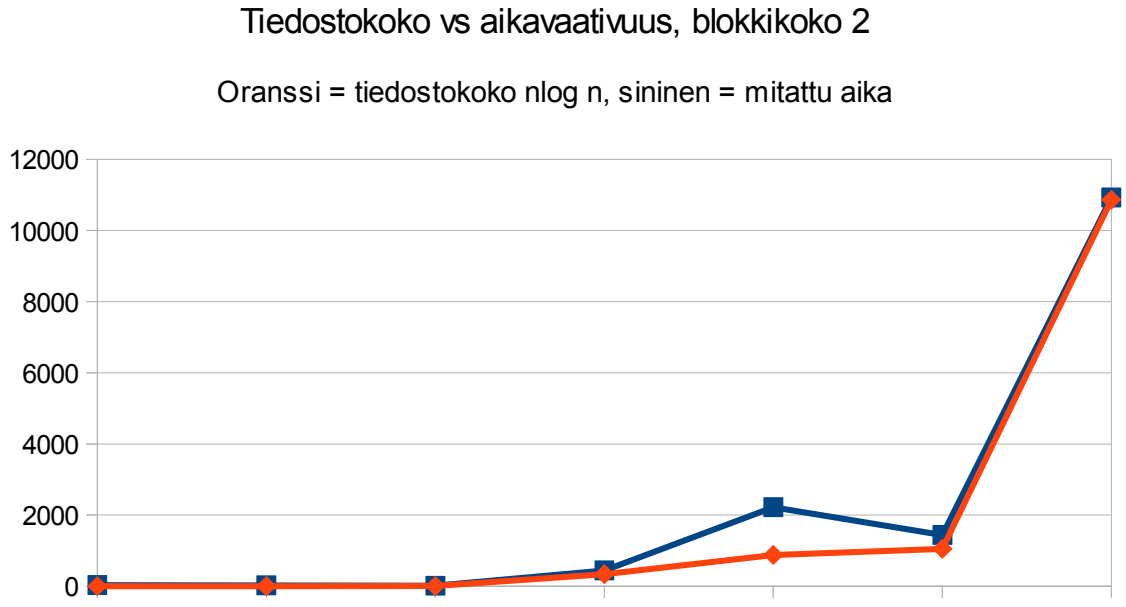

Priority queue - 1000x1000 000 add-operaatiota
ms
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Java: PriorityQueue.add()
Oma: PriorityQueue.add()

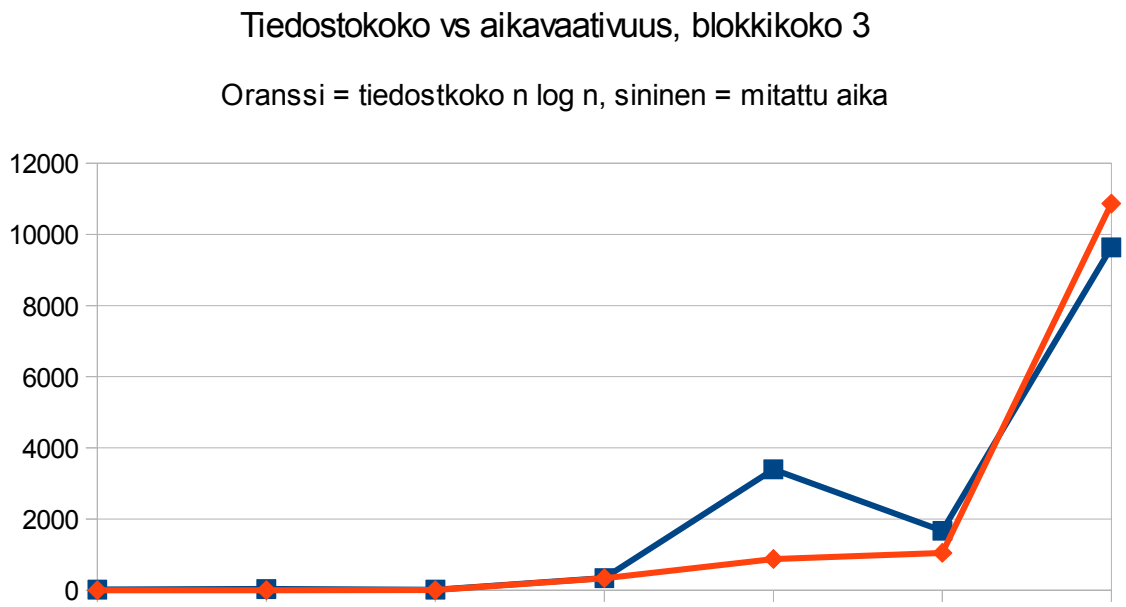

Priority queue 1000x1000 000 poll-operaatiota
ms
30000
25000
20000
15000
10000
5000
0
Java: PriorityQueue.poll()
Oma: PriorityQueue.poll()

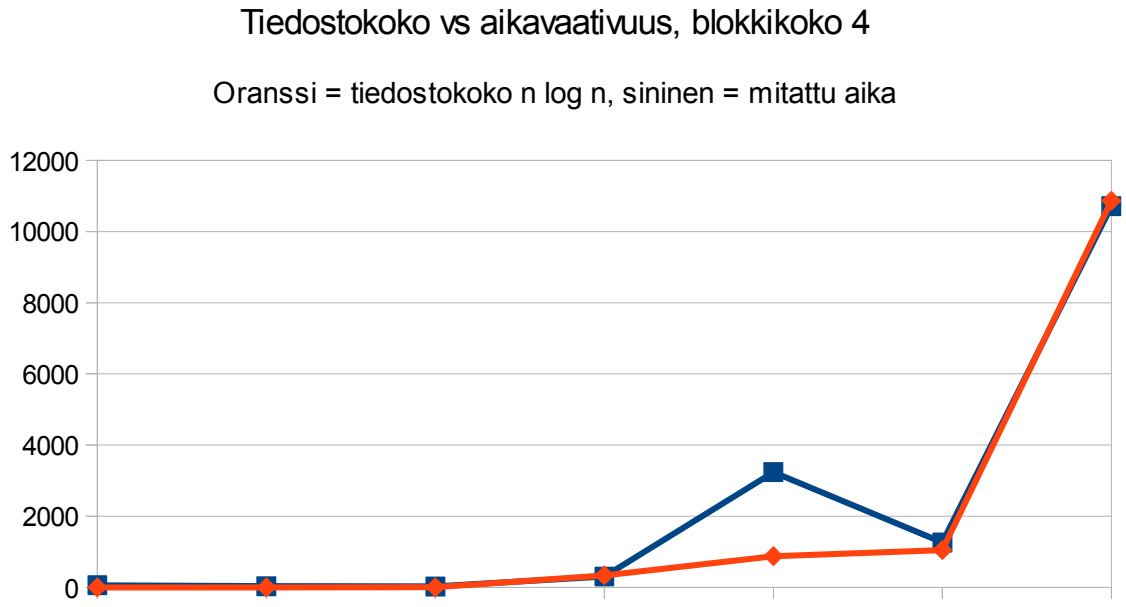
Tiedostokoko vs aikavaativuudet

Käytetyt tiedostot:
tekstitiedosto – 1,6kt
tekstitiedosto – 18kt
tekstitiedosto – 74kt
xml-tiedosto – 2945kt
bmp-tiedosto – 6891kt
bmp-tiedosto – 8100kt
rakenteinen tiedosto, tekstiä – 67710kt







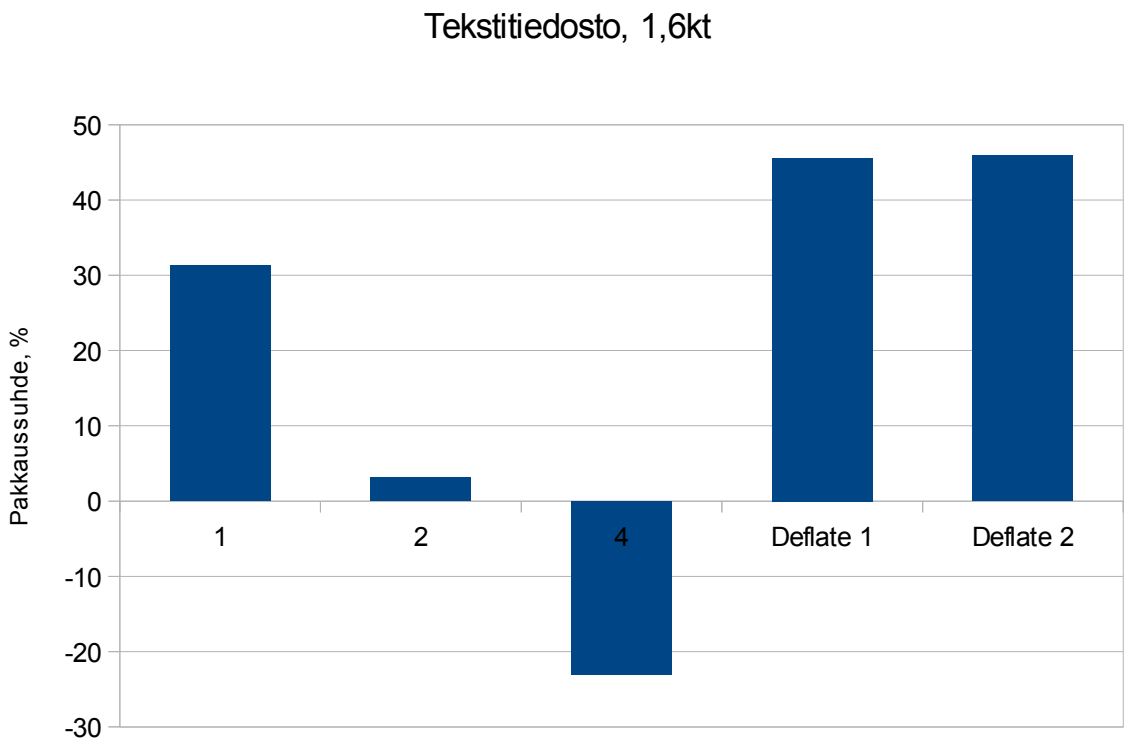


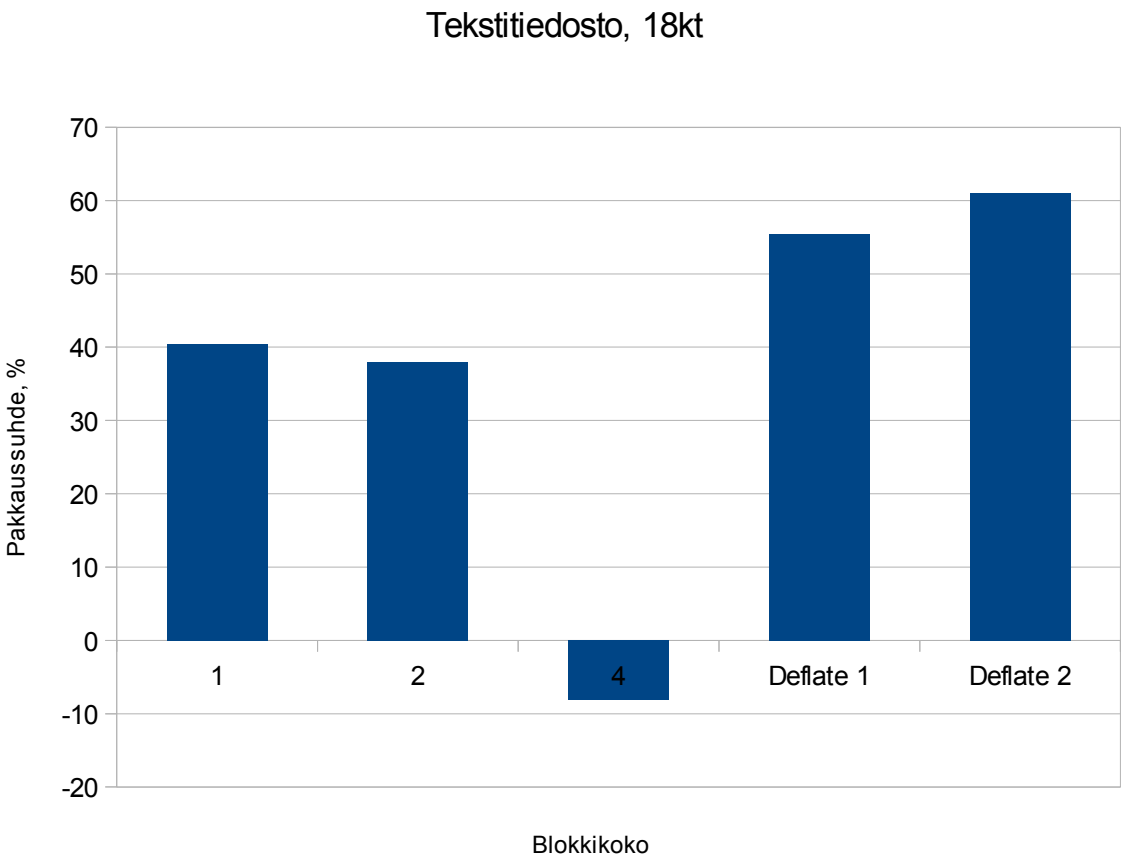
Huom: Tiedostonkoko on skaalattu alaspäin kertoimella 100 jotta käyrät olisivat suunilleen samalla arvovälillä.

Log 2(67000) on noin 16 jolloin logaritminen tekijä enimmäkseen häviää skaalauksessa. Teoreettinen O(n log n)-aikavaativuus vaatii sen että blokkikoko on tarpeeksi suuri että voi generoitua paljon uniikkeja koodeja, ja että tiedoston rakenne on sellainen että koodeja generoituu paljon. Valtaosissa tiedostoja ja pienillä blokkikoilla tätä ei pääse tapahtumaan ja aikavaativuus on noudattaa lähinnä tiedoston kokoa; efektiivisesti O(n) eikä O(n log n) näissä tapauksissa. Kuitenkin yhdellä kuvatiedostolla rakenne on sellainen että O(n log n)-aikavaativuus pääsee dominoimaan koska uniikkeja koodeja generoituu paljon.

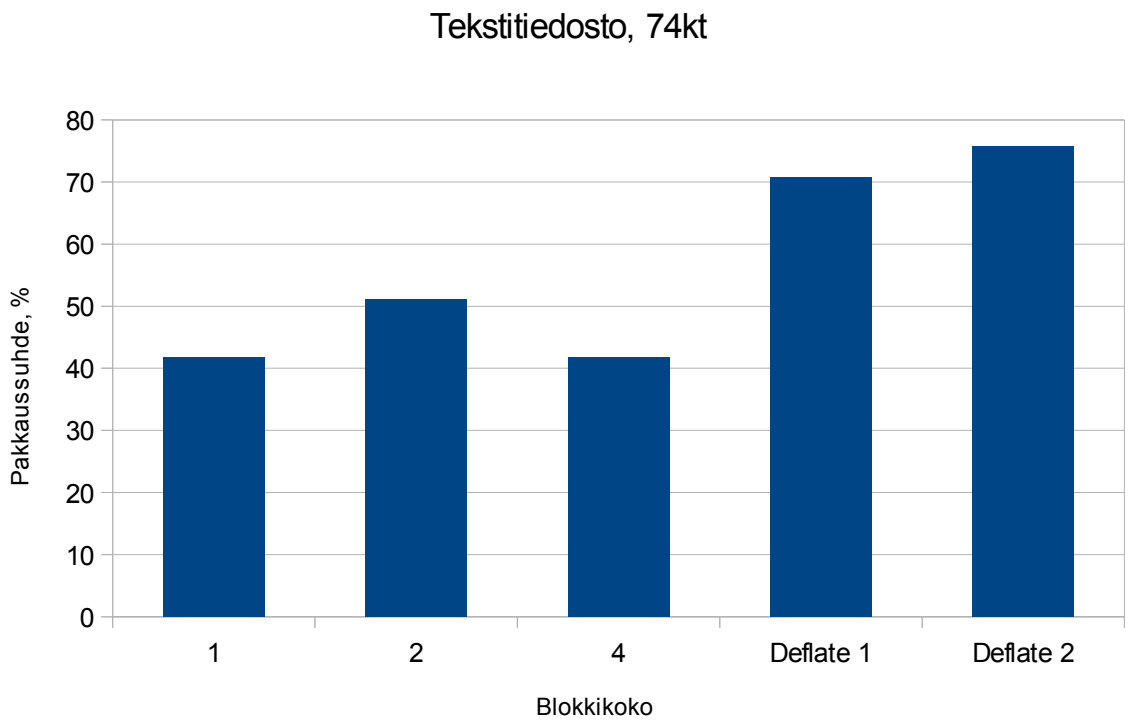
Pakkausprosentit vs blokkikoko ja DEFLATE

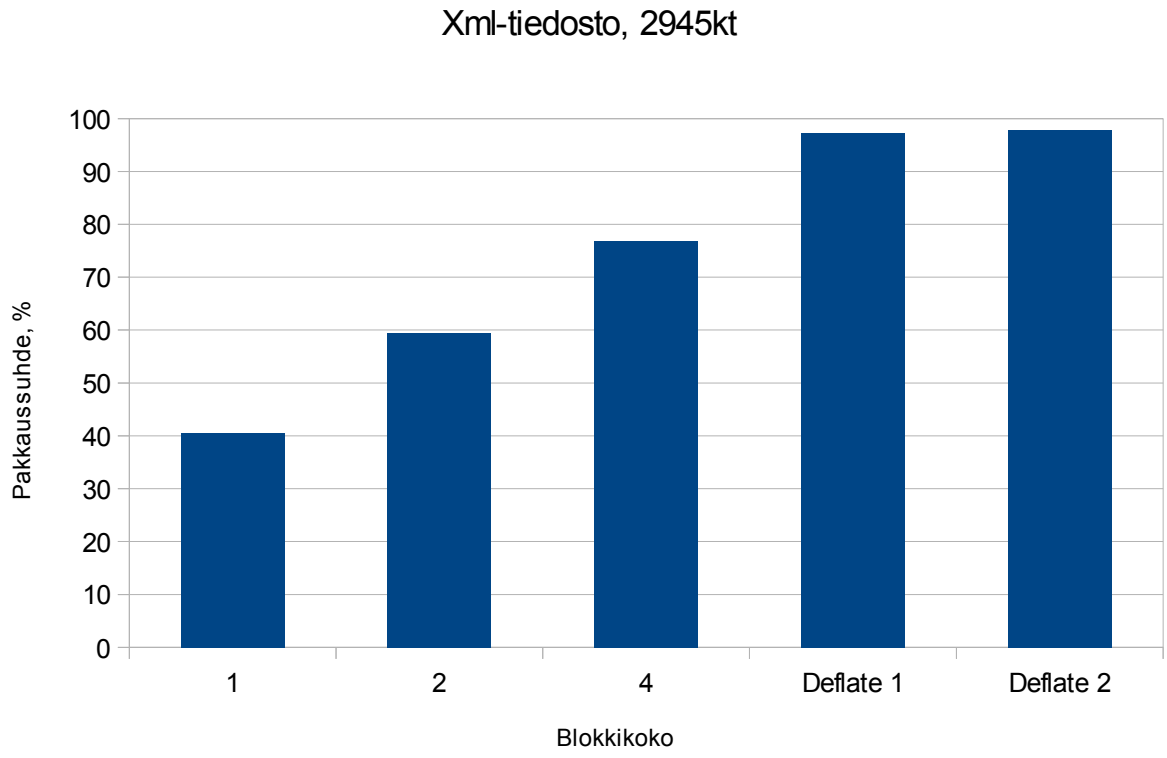
Deflate 1: 7zip, Normaali, 32kb sanakirja, sanan koko 32
Deflate 2: 7zip, Ultra, 32kb sanakirja, sanan koko 128



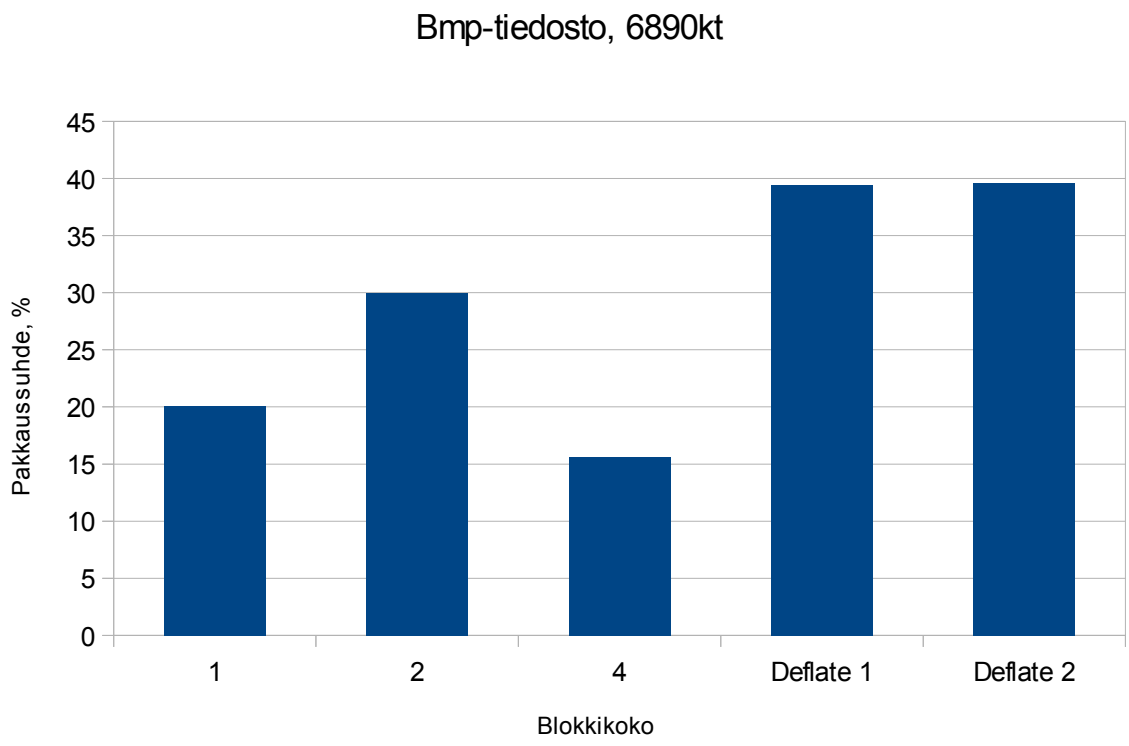


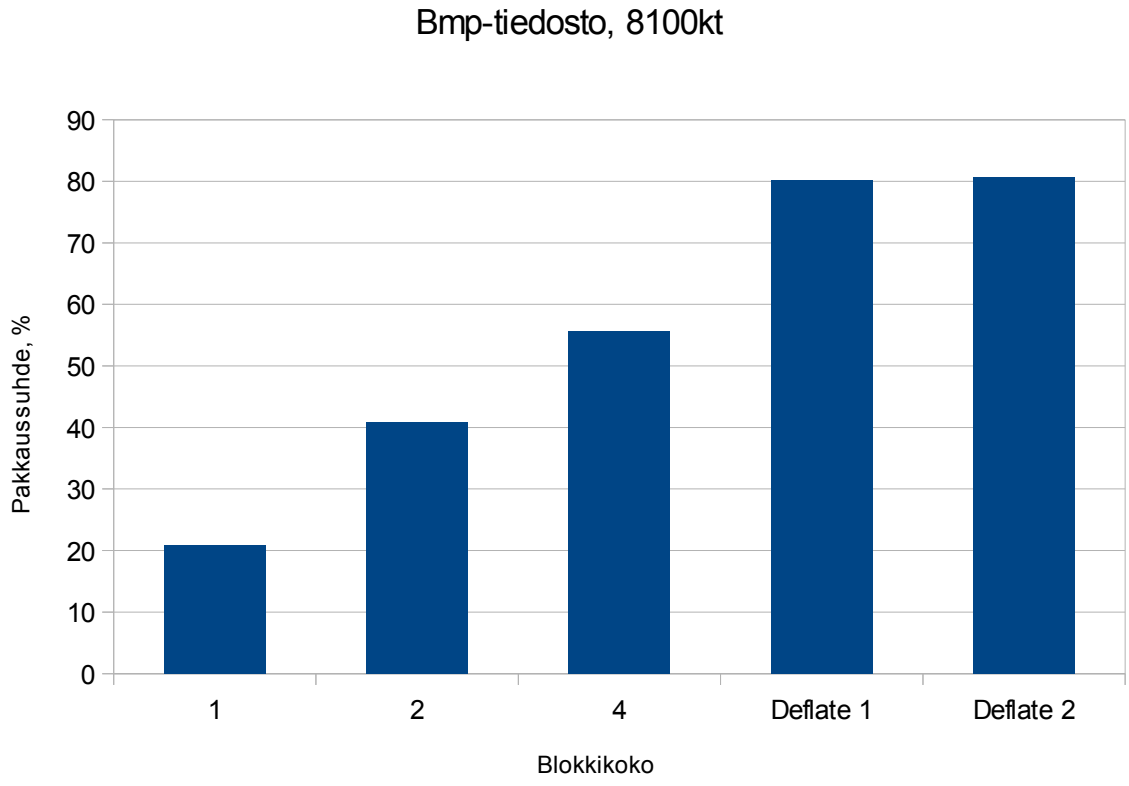
Deflate 1: 7zip, Normaali, 32kb sanakirja, sanan koko 32
Deflate 2: 7zip, Ultra, 32kb sanakirja, sanan koko 128





Deflate 1: 7zip, Normaali, 32kb sanakirja, sanan koko 32
Deflate 2: 7zip, Ultra, 32kb sanakirja, sanan koko 128





Deflate 1: 7zip, Normaali, 32kb sanakirja, sanan koko 32
Deflate 2: 7zip, Ultra, 32kb sanakirja, sanan koko 128