

Testausdokumentti

1 Johdanto

Järjestyslagoritmeja testattiin erikokoisilla syötteillä kolmella eri taulukkotyypillä. Taulukkotyypit olivat **random**, **järjestetty** ja **päinvastainen järjestys**. Ohjelma asetettiin tekemään jokainen järjestäminen 10000 kertaa putkeen ja laskemaan suorituksesta suoritusajan keskiarvo millisekunteina. Kullakin taulukkotyypillä ja -koolla testi toistettiin kolme kertaa, ja tulokseksi laskettiin näiden kolmen mittauksen tuloksen keskiarvo.

Alunperin algoritmeja oli tarkoitus testata 1000, 2000, 5000, 10000 ja 50000 alkion kokoisilla taulukoilla, mutta pian osoittautui järjettömäksi käyttää 50000 alkon taulukkoa, sillä testisuorituksiin alkoi kulua aikaa tunteja. Myöskin tarvittavat johtopäätökset voitiin tehdä pienemmillä taulukoilla. Lopullisien kuvaajien testit tehtiin siis **500, 1000, 2000, 5000, 10000** alkion kokoisilla taulukoilla.

2 Testien mittaustulokset

2.1 Random-taulukko

Random						
Taulukon koko	500	1000	2000	5000	10000	50000
InsertionSort1	0.0573	0.1851	0.7356	3.3897	13.143	332.217
InsertionSort2	0.0582	0.202	0.7111	3.3112	12.9435	315.3816
InsertionSort3	0.0501	0.1788	0.7622	3.4003	12.9265	
avg	0.0552	0.1886	0.7363	3.3671	13.0043	323.7993
MergeSort1	0.0537	0.099	0.2227	0.7396	1.5175	7.3091
MergeSort2	0.0412	0.132	0.2593	0.6452	1.4308	7.9242
MergeSort3	0.0521	0.1064	0.2397	0.7189	1.5245	
avg	0.0490	0.1125	0.2406	0.7012	1.4909	7.6167
QuickSort1	0.0218	0.062	0.1736	0.5023	1.1262	4.9985
QuickSort2	0.0241	0.0827	0.1698	0.4919	1.0861	4.867
QuickSort3	0.0234	0.0698	0.1626	0.4994	1.1335	
avg	0.0231	0.0715	0.1687	0.4979	1.1153	4.9328

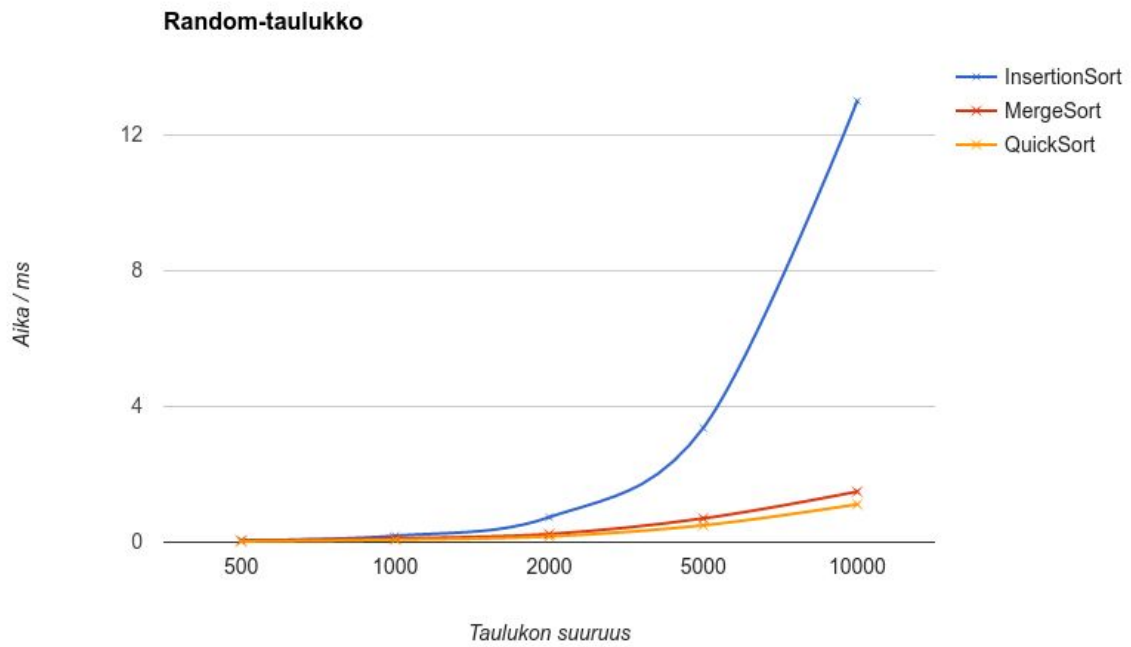
2.2 Järjestetty taulukko

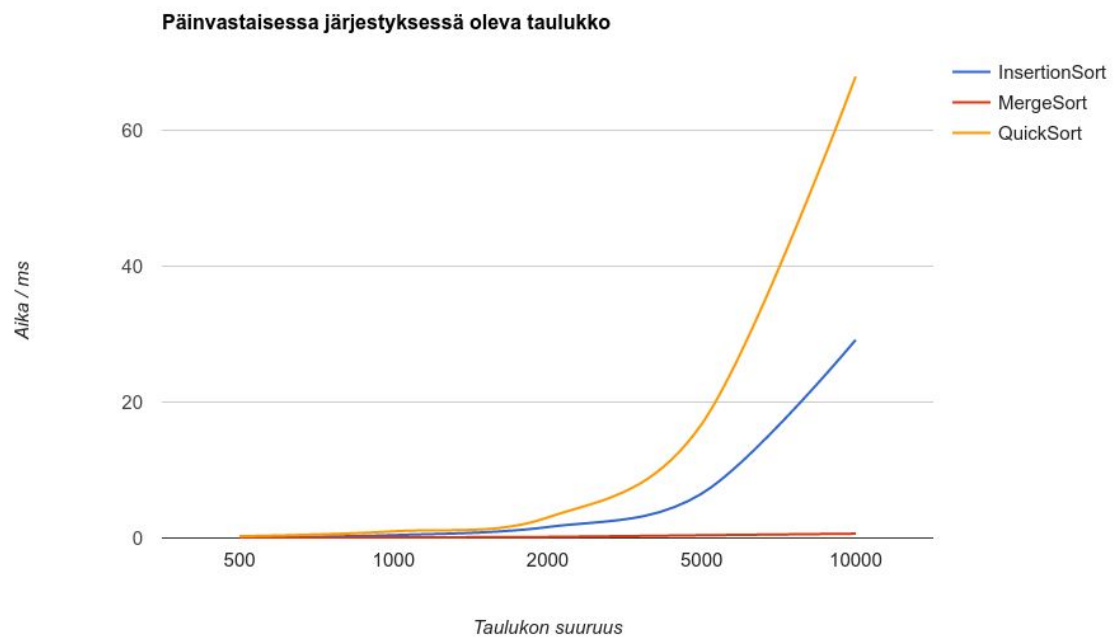
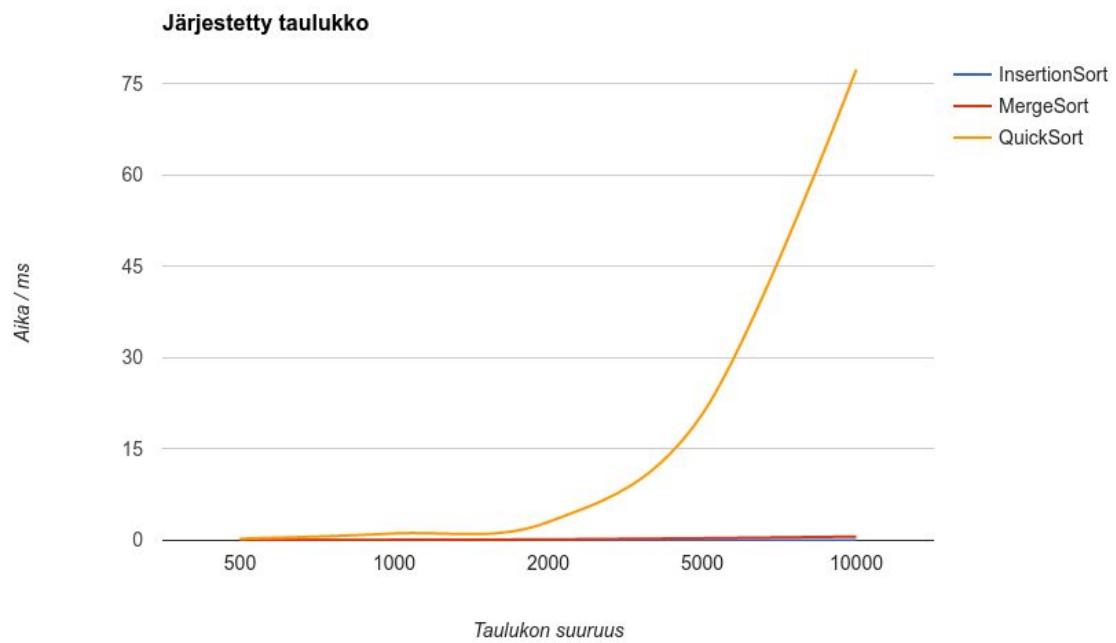
Ordered					
Taulukon koko	500	1000	2000	5000	10000
InsertionSort1	0.001	0.0022	0.0023	0.0047	0.0083
InsertionSort2	0.0013	0.0015	0.0022	0.0045	0.0033
InsertionSort3	7.00E-04	0.0019	0.0034	0.0042	0.0057
avg	0.0010	0.0019	0.0026	0.0045	0.0058
MergeSort1	0.0623	0.0754	0.1543	0.3691	0.7597
MergeSort2	0.038	0.0829	0.1819	0.4024	0.5104
MergeSort3	0.0557	0.0959	0.1431	0.3187	0.6317
avg	0.0520	0.0847	0.1598	0.3634	0.6339
QuickSort1	0.2981	1.1657	2.7982	20.8228	83.7523
QuickSort2	0.2431	1.1433	3.4356	20.8301	81.917
QuickSort3	0.3155	1.1412	2.815	20.5352	66.4487
avg	0.2856	1.1501	3.0163	20.7294	77.3727

2.3 Päinvastaisessa järjestyksessä oleva taulukko

Reverse order					
Taulukon koko	500	1000	2000	5000	10000
InsertionSort1	0.1129	0.407	1.6522	6.5342	29.2331
InsertionSort2	0.1069	0.4346	1.5216	6.5404	32.8359
InsertionSort3	0.1009	0.4362	1.667	6.5297	25.5398
avg	0.1069	0.4259	1.6136	6.5348	29.2029
MergeSort1	0.0402	0.092	0.156	0.4	0.7336
MergeSort2	0.0355	0.0926	0.1705	0.4079	0.5689
MergeSort3	0.0368	0.0746	0.1588	0.4125	0.6116
avg	0.0375	0.0864	0.1618	0.4068	0.6380
QuickSort1	0.2818	1.0236	3.2616	16.7622	72.84
QuickSort2	0.2401	0.9684	2.927	16.8068	65.4971
QuickSort3	0.2496	0.9672	2.8593	16.7579	65.6653
avg	0.2572	0.9864	3.0160	16.7756	68.0008

3 Tulosten kuvaajat





4 Johtopäätökset mittauksista

4.1 Random taulukoiden testit

Tuloksista nähdään, että sekä QuickSort että MergeSort suorittavat järjestämisen taulukon koosta riippumatta ajassa $O(n \log n)$. InsertionSort sen sijaan hidastuu taulukon kasvaessa huomattavasti, ja sen järjestämiseen käyttämä aika kasvaa eksponentiaalisesti taulukon koon kasvaessa. Näin sen aikavaativuus on ennakoitu $O(n^2)$

4.2 Järjestetyn taulukon testit

Järjestetyssä taulukossa tulokset viittaavat siihen, että MergeSort ja InsertionSort suoriutuvat järjestämisestä lineaarisessa ajassa $O(n)$, mutta QuickSortin aikavaativuus on $O(n^2)$. QuickSort suoriutuu keskimäärin ajassa $O(n \log n)$, mutta tapauksissa joissa jokaisella kierroksella asetetun sarana-alkion vasemmalle (tai oikealle) puolelle ei siirretä yhtään alkioita on sen aikavaativuus huonoin, eli $O(n^2)$. Omassa algoritmissani tämä johtuu siitä, että sarana-alkio on määritetty aina taulukon ensimmäiseksi alkikoksi. Mikäli se olisi taulukon mediaani, ei järjestetyn taulukon ongelmaa ilmenisi.

4.3 Päinvastaisessa järjestyksessä olevan taulukon testit.

Tässäkin tapauksessa MergeSort suoriutuu järjestämisestä lineaarisessa ajassa. QuickSort vaatii aikaa $O(n^2)$, mutta se olisi korjattavissa kohdan 4.2 mukaisella ratkaisulla. InsertionSort suoriutuu tästä taulukkotyypistä tehokkaammin kuin random-tilukosta. Sen aikavaativuus vaikuttaisi tässä tapauksessa olevan $O(1/n \log n)$.