

Verslag Practicum 1:

Arnout Coenegrachts, r0665757

27 maart 2020

De input data is telkens een willekeurig gegenereerde lijst met integers van 0 tot en met 99.

1 Aantal vergelijkingen

In dit eerste deel van het practicum, zijn het aantal vergelijkingen tussen 2 elementen van een lijst geteld bij 3 verschillende sorteeralgoritmen, namelijk selection sort, insertion sort en quick sort. Voor elk sorteeralgoritmen zijn er lijsten met een lengte van 1 tot en met 100 elementen gesorteerd. Voor elke lengte zijn er 3 verschillende lijsten gegenereerd, waardoor elk sorteeralgoritme dus 300 lijsten gesorteerd heeft. Een grafiek met de data van de 3 verschillende sorteeralgoritmen kan gezien worden in figuur 4 op het einde van deze sectie.

1.1 Selection Sort

Omdat selection sort telkens het volledige ongesorteerde deel van de lijst afloopt om het minimum te vinden, zou het aantal vergelijkingen gelijk moeten blijven voor elke lijst van dezelfde grootte, ongeacht de inhoud. Het verwachte aantal vergelijkingen wordt gegeven door $N(N-1)/2$ of $\sim N^2/2$. In figuur 1 is het $\sim N^2/2$ model geplot in het blauw en de waardes bekomen door de experimenten geplot met kruisjes in het rood, paars en licht blauw. Op deze figuur is duidelijk te zien dat het model zoals verwacht goed overeenkomt met de waardes van de experimenten, en dat de werkelijke waardes voor de 3 experimenten exact hetzelfde zijn. De data staat in de appendix in tabel 1.

1.2 Insertion Sort

Bij insertion sort zijn er 3 gevallen te onderscheiden. Het slechtste geval is wanneer de lijst van groot naar klein gesorteerd is waardoor de kleinere elementen van achteraan helemaal naar voor moeten verplaatst worden, wat veel vergelijkingen oplevert. In dit geval verwacht men $\sim N^2/2$, wat even veel vergelijkingen zijn als bij selection sort. In het beste geval is de lijst al gesorteerd van klein naar groot, waardoor er $N-1$ of $\sim N$ vergelijkingen nodig zijn, namelijk om de volledige lijst af te lopen. In het gemiddelde geval verwacht men $\sim N^2/4$ vergelijkingen, omdat elk element gemiddeld tot in de helft van de lijst verplaatst moet worden. Deze drie gevallen zijn geplot in figuur 2, samen met de waardes van de experimenten. Het

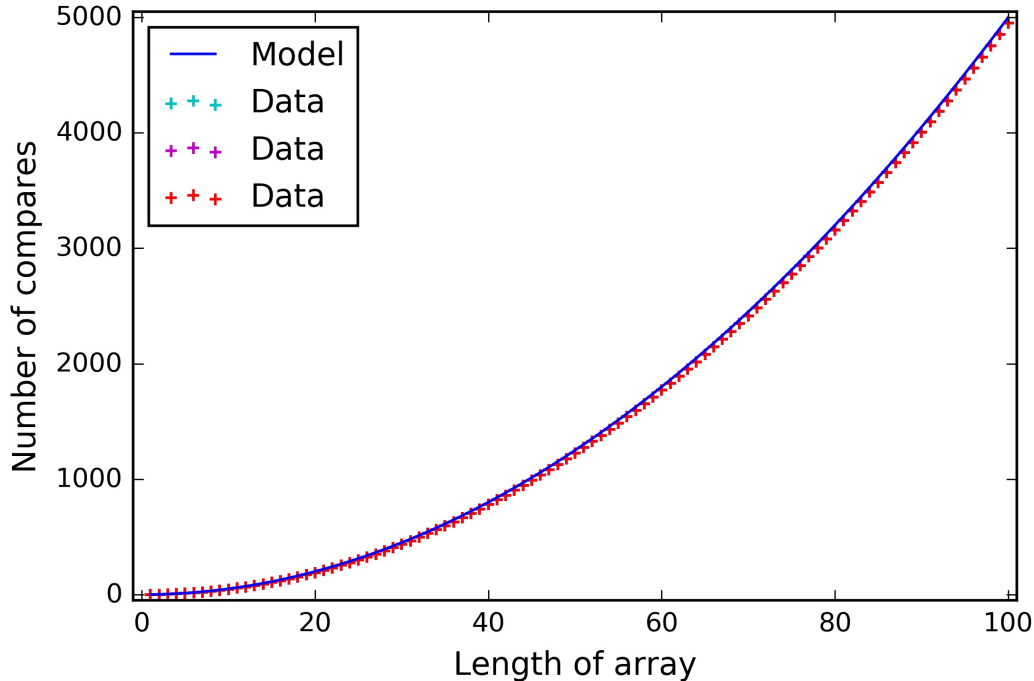


Figure 1: *Het aantal vergelijkingen tussen 2 elementen van een lijst bij het sorteren via selection sort.*

slechtste geval, beste geval en gemiddelde geval zijn geplot met een rode, groene en blauwe lijn respectievelijk. De bekomen data is geplot in rode, paarse en licht blauwe kruisjes. We kunnen op de figuur duidelijk zien dat alle data geconcentreerd is rond het gemiddelde geval, en dat het bij de grotere waarden steeds verder van de 2 extreme gevallen wegblijft. Dit is ook logisch, omdat het aantal elementen dat al op de juiste plaats staat of dat naar de andere kant van de lijst moet gaan heel klein gaat zijn. De data staat in de appendix in tabel 2.

1.3 Quick Sort

Bij quick sort zijn er ook 3 gevallen te onderscheiden. Het slechtste geval is wanneer het pivot telkens het grootste of het kleinste element is van de lijst, waardoor alle elementen terug in 1 lijst komen te staan. In dit geval verwacht men $\sim N^2/2$ vergelijkingen. In het beste geval wordt de lijst telkens in 2 gedeeld, en dan verwacht men $\sim N \log_2(N)$ vergelijkingen. In het gemiddelde geval verwacht men $\sim 1.39 N \log_2(N)$. Deze drie gevallen zijn geplot in figuur 3, samen met de waarden van de experimenten. Het slechtste geval, beste geval en gemiddelde geval zijn geplot met een rode, groene en blauwe lijn respectievelijk. De bekomen data is geplot in rode,

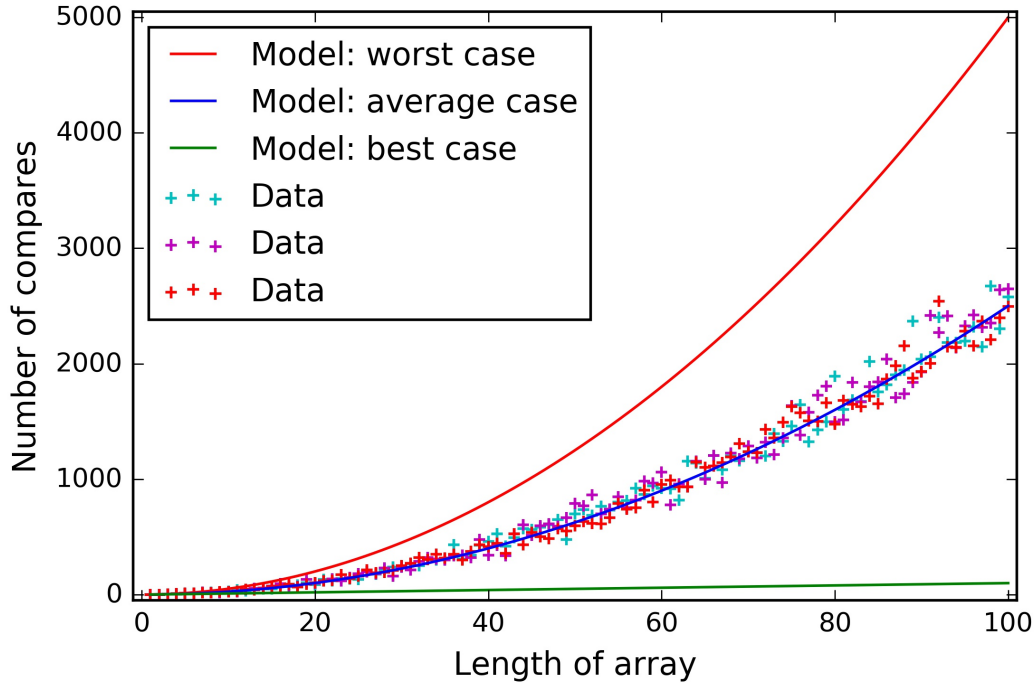


Figure 2: *Het aantal vergelijkingen tussen 2 elementen van een lijst bij het sorteren via insertion sort.*

paarse en licht blauwe kruisjes. Op de figuur is te zien dat de data geconcentreerd is vlak boven het gemiddelde geval, wat aangeeft dat het slechtste geval zich een significant aantal keer meer heeft voor gedaan dan het beste geval. Tegen dit slechtste geval kan men zich beschermen door de lijst telkens willekeurig van volgorde te laten wisselen wanneer er een nieuwe partitie gemaakt wordt, maar dit is niet gedaan omdat anders het aantal vergelijkingen dat nodig is om eenzelfde lijst te sorteren niet altijd hetzelfde zou zijn, wat een vereiste was van dit practicum. De data staat in de appendix in tabel 3.

2 Doubling Ratio

In dit deel van het practicum, wordt de hoeveelheid tijd die er nodig is om elk sorteer algoritme uit te voeren gemeten, terwijl de lengte van de lijst die gesorteerd wordt telkens verdubbelt wordt. De tijdsmeting wordt gebeurt door nanoTime van Java. We passen om 2 opeenvolgende metingen telkens de volgende formule toe:

$$\frac{T(2N)}{T(N)} \sim 2^b \quad (1)$$

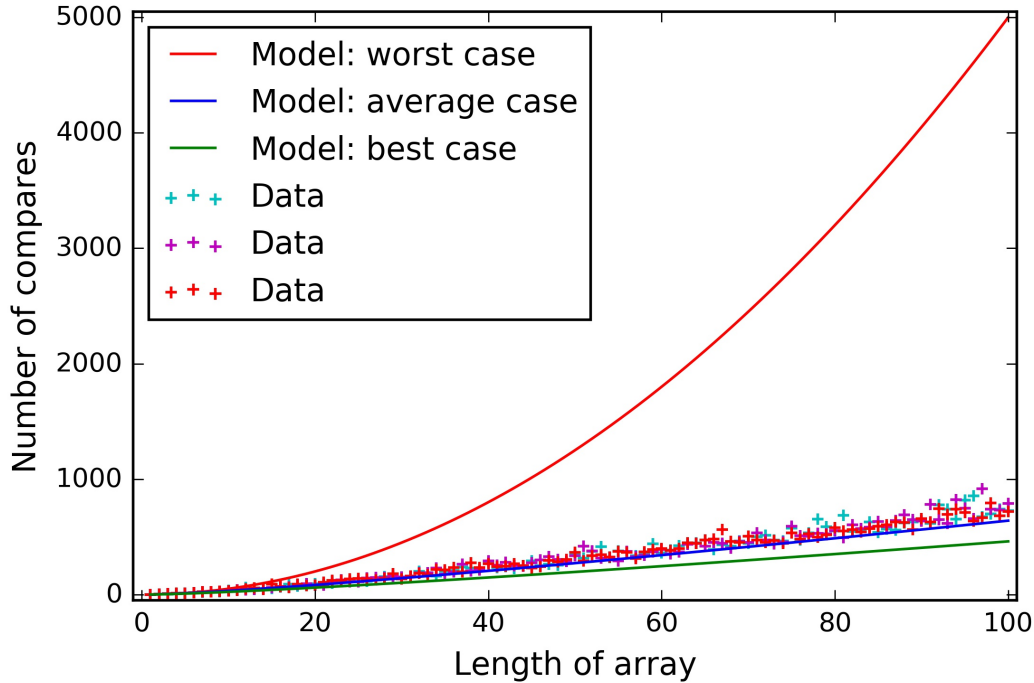


Figure 3: *Het aantal vergelijkingen tussen 2 elementen van een lijst bij het sorteren via quick sort.*

Deze formule zegt ons dat deze verhouding voor grote N gaat convergeren naar een macht van 2. Hierdoor kunnen we de looptijd van nog grotere lijsten trachten te voorspellen.

2.1 Insertion Sort

Aangezien bij het gemiddelde geval het aantal vergelijkingen bij insertion sort gegeven wordt door $\sim N^2/4$, geeft de formule van de doubling ratio (1) aan dat de verhouding zal convergeren naar $4 = 2^2$. In figuur 5 is het zichtbaar dat vanaf een lijst-grootte van 10^4 elementen, de verhouding rond 4 blijft. De data staat in de appendix in tabel 4. De grootste meting hiervan had een lengte van $N = 128000$ en duurde 1.756×10^{10} nanoseconden, ofwel 17.56 seconden. Om de tijd van een meting met een 8 keer zo lange rij te bepalen, gebruiken we de volgende vergelijking:

$$\frac{T(8N)}{T(N)} = \frac{T(8N)}{T(4N)} \times \frac{T(4N)}{T(2N)} \times \frac{T(2N)}{T(N)} \quad (2)$$

Voor insertion sort is de oplossing hiervan $2^6 = 64$. De sortering van een lijst van $N = 1024000$ zou dus $2^6 \times 17.56 = 1123.84$ seconden duren, wat ongeveer 19 minuten is.

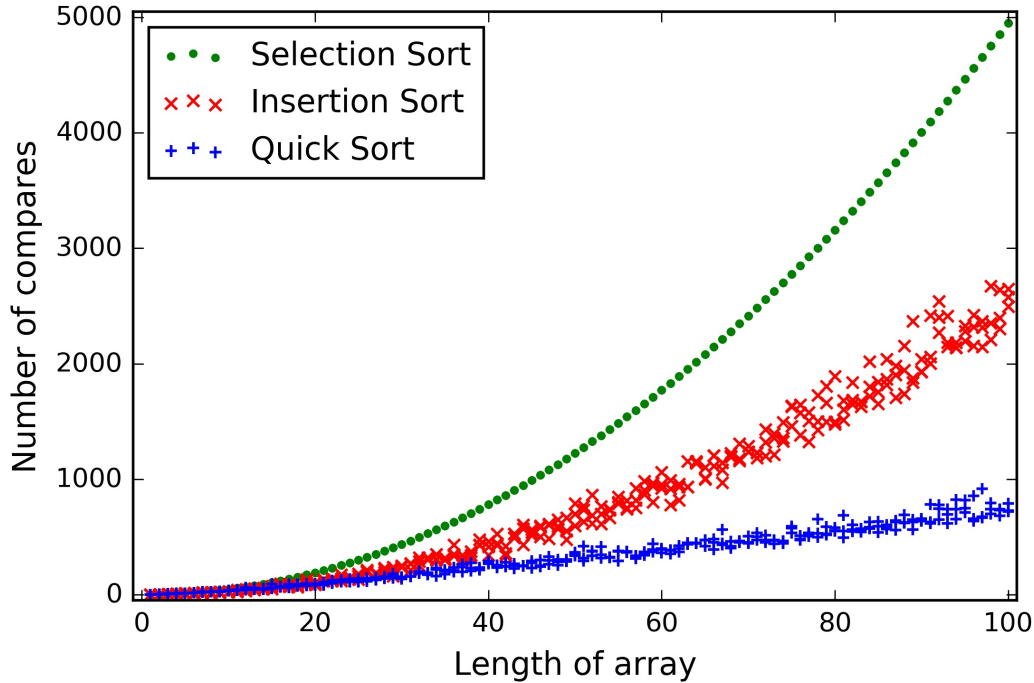


Figure 4: De data van de 3 verschillende sorteer algoritmes. De data van selection sort, insertion sort en quick sort zijn geplot als groene bolletjes; rode x-en; en blauwe kruisjes respectievelijk

2.2 Quick Sort

Aangezien bij het gemiddelde geval het aantal vergelijkingen bij quick sort gegeven wordt door $\sim 1.39N \log_2(N)$, geeft de formule (1) de volgende oplossing:

$$\frac{T(2N)}{T(N)} = \frac{1.39 \times 2N \log_2(2N)}{1.39 \times N \log_2(N)} = \frac{2N(1 + \log_2(N))}{N \log_2(N)} = 2 \times \left(1 + \frac{1}{\log_2(N)}\right)$$

We verwachten dus dat de verhouding zal convergeren naar een getal net groter dan 2, want voor grote N wordt de term $1/\log_2(N)$ heel klein. In figuur 6 is het duidelijk dat vanaf een lijst grootte van 10^6 elementen, de verhouding net boven 2 blijft. De data staat in de appendix in tabel 5. De grootste meting hiervan had een lengte van $N = 65536000$ en duurde 1.588×10^{10} nanoseconden, ofwel 15.88 seconden. Om de tijd van een meting met een 8 keer zo lange rij te bepalen, gebruiken we opnieuw vergelijking (2). Voor quick sort is de oplossing van deze vergelijking $2^3 = 8$. De sortering van een lijst van $N = 524288000$ zou dus $2^3 \times 15.88 = 127.04$ seconden duren, wat ongeveer 2 minuten is.

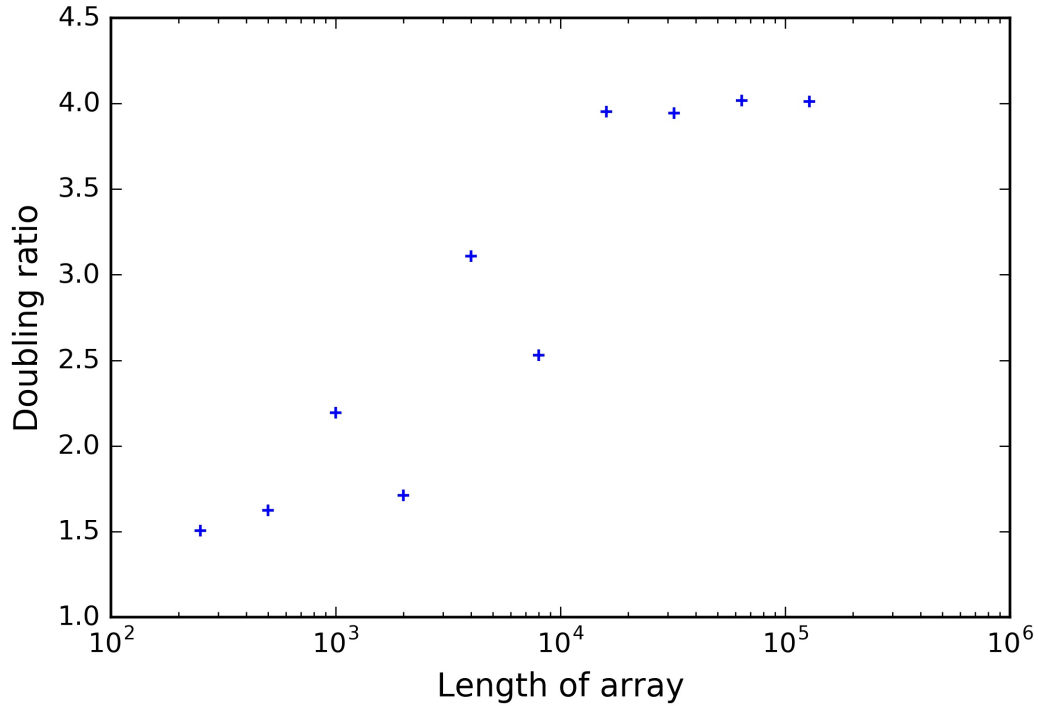


Figure 5: *De data van het doubling ratio experiment voor insertion sort*

2.3 Algoritme met $\sim N^5$

Voor een algoritme met $\sim N^5$, wordt de doubling ratio berekend als volgt:

$$\frac{T(2N)}{T(N)} = \frac{(2N)^5}{N^5} = 2^5 = 32$$

Als de invoer dus 2 keer zou lang zou zijn, dan verwachten we een tijdsduur die 32 keer zo lang is.

3 Appendix

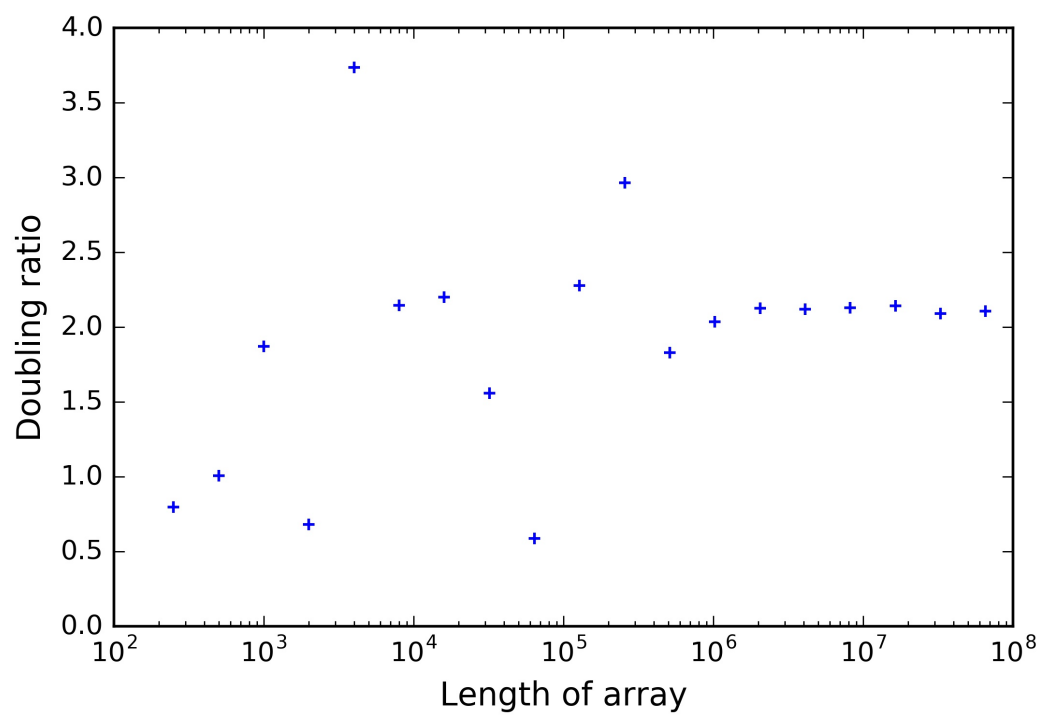


Figure 6: *De data van het doubling ratio experiment voor insertion sort*

Table 1: Aantal vergelijkingen voor selection sort voor lijsten met een lengte tussen 1 en 100 elementen

N	1	2	3	4	5	6	7	8	9	10
<i>Experiment1</i>	0	1	3	6	10	15	21	28	36	45
<i>Experiment2</i>	0	1	3	6	10	15	21	28	36	45
<i>Experiment3</i>	0	1	3	6	10	15	21	28	36	45
N	11	12	13	14	15	16	17	18	19	20
<i>Experiment1</i>	55	66	78	91	105	120	136	153	171	190
<i>Experiment2</i>	55	66	78	91	105	120	136	153	171	190
<i>Experiment3</i>	55	66	78	91	105	120	136	153	171	190
N	21	22	23	24	25	26	27	28	29	30
<i>Experiment1</i>	210	231	253	276	300	325	351	378	406	435
<i>Experiment2</i>	210	231	253	276	300	325	351	378	406	435
<i>Experiment3</i>	210	231	253	276	300	325	351	378	406	435
N	31	32	33	34	35	36	37	38	39	40
<i>Experiment1</i>	465	496	528	561	595	630	666	703	741	780
<i>Experiment2</i>	465	496	528	561	595	630	666	703	741	780
<i>Experiment3</i>	465	496	528	561	595	630	666	703	741	780
N	41	42	43	44	45	46	47	48	49	50
<i>Experiment1</i>	820	861	903	946	990	1035	1081	1128	1176	1225
<i>Experiment2</i>	820	861	903	946	990	1035	1081	1128	1176	1225
<i>Experiment3</i>	820	861	903	946	990	1035	1081	1128	1176	1225
N	51	52	53	54	55	56	57	58	59	60
<i>Experiment1</i>	1275	1326	1378	1431	1485	1540	1596	1653	1711	1770
<i>Experiment2</i>	1275	1326	1378	1431	1485	1540	1596	1653	1711	1770
<i>Experiment3</i>	1275	1326	1378	1431	1485	1540	1596	1653	1711	1770
N	61	62	63	64	65	66	67	68	69	70
<i>Experiment1</i>	1830	1891	1953	2016	2080	2145	2211	2278	2346	2415
<i>Experiment2</i>	1830	1891	1953	2016	2080	2145	2211	2278	2346	2415
<i>Experiment3</i>	1830	1891	1953	2016	2080	2145	2211	2278	2346	2415
N	71	72	73	74	75	76	77	78	79	80
<i>Experiment1</i>	2485	2556	2628	2701	2775	2850	2926	3003	3081	3160
<i>Experiment2</i>	2485	2556	2628	2701	2775	2850	2926	3003	3081	3160
<i>Experiment3</i>	2485	2556	2628	2701	2775	2850	2926	3003	3081	3160
N	81	82	83	84	85	86	87	88	89	90
<i>Experiment1</i>	3240	3321	3403	3486	3570	3655	3741	3828	3916	4005
<i>Experiment2</i>	3240	3321	3403	3486	3570	3655	3741	3828	3916	4005
<i>Experiment3</i>	3240	3321	3403	3486	3570	3655	3741	3828	3916	4005
N	91	92	93	94	95	96	97	98	99	100
<i>Experiment1</i>	4095	4186	4278	4371	4465	4560	4656	4753	4851	4950
<i>Experiment2</i>	4095	4186	4278	4371	4465	4560	4656	4753	4851	4950
<i>Experiment3</i>	4095	4186	4278	4371	4465	4560	4656	4753	4851	4950

Table 2: Aantal vergelijkingen voor insertion sort voor lijsten met een lengte tussen 1 en 100 elementen

N	1	2	3	4	5	6	7	8	9	10
<i>Experiment1</i>	0	1	3	5	7	11	15	18	20	30
<i>Experiment2</i>	0	1	2	5	8	9	18	17	14	24
<i>Experiment3</i>	0	1	2	6	10	12	17	15	25	25
N	11	12	13	14	15	16	17	18	19	20
<i>Experiment1</i>	40	34	48	50	47	78	70	82	104	86
<i>Experiment2</i>	25	44	54	58	72	94	72	71	130	100
<i>Experiment3</i>	28	43	40	51	69	73	88	66	91	104
N	21	22	23	24	25	26	27	28	29	30
<i>Experiment1</i>	133	119	139	137	130	199	177	216	238	248
<i>Experiment2</i>	117	135	118	148	179	195	180	228	161	238
<i>Experiment3</i>	118	123	173	141	159	212	189	193	208	251
N	31	32	33	34	35	36	37	38	39	40
<i>Experiment1</i>	266	251	302	298	308	431	308	341	429	458
<i>Experiment2</i>	213	287	321	309	300	336	341	320	477	340
<i>Experiment3</i>	272	322	304	350	318	348	301	372	433	416
N	41	42	43	44	45	46	47	48	49	50
<i>Experiment1</i>	527	419	493	571	563	584	604	649	476	700
<i>Experiment2</i>	429	337	525	603	513	595	612	592	664	791
<i>Experiment3</i>	445	359	525	432	539	503	485	573	551	596
N	51	52	53	54	55	56	57	58	59	60
<i>Experiment1</i>	736	688	763	733	803	814	920	866	940	920
<i>Experiment2</i>	767	864	699	740	846	752	819	981	965	1059
<i>Experiment3</i>	631	618	614	664	787	741	750	904	803	953
N	61	62	63	64	65	66	67	68	69	70
<i>Experiment1</i>	917	819	1155	1139	1005	1204	1081	1201	1161	1236
<i>Experiment2</i>	777	957	932	1155	999	1203	969	1224	1174	1288
<i>Experiment3</i>	991	931	933	1146	1101	1114	1143	1192	1308	1239
N	71	72	73	74	75	76	77	78	79	80
<i>Experiment1</i>	1229	1202	1394	1328	1460	1645	1324	1426	1497	1891
<i>Experiment2</i>	1184	1319	1212	1357	1636	1380	1579	1727	1806	1496
<i>Experiment3</i>	1230	1430	1358	1494	1629	1573	1505	1500	1661	1475
N	81	82	83	84	85	86	87	88	89	90
<i>Experiment1</i>	1602	1686	1675	2019	1757	1817	1905	1943	2370	2040
<i>Experiment2</i>	1512	1839	1671	1799	1840	2040	1707	1738	1837	1929
<i>Experiment3</i>	1680	1649	1629	1717	1651	1866	1980	2156	1874	1931
N	91	92	93	94	95	96	97	98	99	100
<i>Experiment1</i>	2060	2403	2184	2182	2196	2320	2146	2671	2304	2579
<i>Experiment2</i>	2416	2271	2413	2137	2328	2420	2315	2353	2638	2647
<i>Experiment3</i>	2001	2542	2147	2142	2280	2154	2367	2207	2399	2495

Table 3: Aantal vergelijkingen voor quick sort voor lijsten met een lengte tussen 1 en 100 elementen

N	1	2	3	4	5	6	7	8	9	10
<i>Experiment1</i>	0	3	4	9	10	15	20	24	26	35
<i>Experiment2</i>	0	3	6	7	11	16	25	23	27	34
<i>Experiment3</i>	0	3	6	11	10	14	22	23	29	33
N	11	12	13	14	15	16	17	18	19	20
<i>Experiment1</i>	39	61	43	51	53	65	91	72	88	95
<i>Experiment2</i>	39	41	43	50	59	70	67	84	92	83
<i>Experiment3</i>	35	47	55	45	90	69	63	88	76	81
N	21	22	23	24	25	26	27	28	29	30
<i>Experiment1</i>	90	98	109	120	113	115	132	155	143	150
<i>Experiment2</i>	81	115	128	112	110	143	149	146	158	151
<i>Experiment3</i>	107	124	122	134	138	136	126	141	182	136
N	31	32	33	34	35	36	37	38	39	40
<i>Experiment1</i>	154	199	189	158	201	208	239	228	223	278
<i>Experiment2</i>	161	188	187	230	177	195	262	206	212	292
<i>Experiment3</i>	161	186	172	219	212	233	196	273	233	268
N	41	42	43	44	45	46	47	48	49	50
<i>Experiment1</i>	235	244	225	250	291	252	262	255	287	343
<i>Experiment2</i>	253	281	251	247	275	299	330	304	289	338
<i>Experiment3</i>	240	246	268	249	230	240	294	282	305	364
N	51	52	53	54	55	56	57	58	59	60
<i>Experiment1</i>	314	372	413	319	377	369	336	359	440	361
<i>Experiment2</i>	417	378	317	324	292	374	327	341	369	392
<i>Experiment3</i>	290	335	346	324	375	367	312	368	389	397
N	61	62	63	64	65	66	67	68	69	70
<i>Experiment1</i>	388	423	449	446	471	389	432	420	465	445
<i>Experiment2</i>	355	368	437	444	420	453	442	401	460	453
<i>Experiment3</i>	382	399	446	445	474	482	562	459	453	505
N	71	72	73	74	75	76	77	78	79	80
<i>Experiment1</i>	503	514	472	466	576	473	535	652	587	549
<i>Experiment2</i>	534	473	441	455	590	510	504	529	525	554
<i>Experiment3</i>	473	453	467	442	535	480	526	497	516	581
N	81	82	83	84	85	86	87	88	89	90
<i>Experiment1</i>	687	545	556	629	539	568	557	621	629	643
<i>Experiment2</i>	494	603	571	585	632	589	639	692	650	636
<i>Experiment3</i>	549	564	543	571	593	603	629	623	562	656
N	91	92	93	94	95	96	97	98	99	100
<i>Experiment1</i>	618	776	739	655	816	853	675	700	739	726
<i>Experiment2</i>	781	651	617	822	748	658	916	741	732	787
<i>Experiment3</i>	628	742	694	740	711	635	666	792	682	721

Table 4: De tijd die nodig is om insertion sort uit te voeren bij verdubbende lijst-grootte en de bijhorende doubling ratio.

N	$T(N)[ns]$	$T(2N)/T(N)$
250	1.457×10^6	1.505
500	2.366×10^6	1.623
1000	5.190×10^6	2.193
2000	8.883×10^6	1.712
4000	2.760×10^7	3.107
8000	6.986×10^7	2.531
16000	2.761×10^8	3.952
32000	1.089×10^9	3.945
64000	4.377×10^9	4.018
128000	1.756×10^{10}	4.012

Table 5: De tijd die nodig is om quick sort uit te voeren bij verdubbende lijst-grootte en de bijhorende doubling ratio.

N	$T(N)[ns]$	$T(2N)/T(N)$
250	3.371×10^5	0.795
500	3.397×10^5	1.008
1000	6.352×10^5	1.87
2000	4.333×10^5	0.682
4000	1.618×10^6	3.734
8000	3.471×10^6	2.145
16000	7.633×10^6	2.199
32000	1.189×10^7	1.557
64000	6.977×10^6	0.587
128000	1.589×10^7	2.278
256000	4.712×10^7	2.964
512000	8.615×10^7	1.828
1024000	1.755×10^8	2.037
2048000	3.731×10^8	2.126
4096000	7.905×10^8	2.119
8192000	1.684×10^9	2.13
16384000	3.604×10^9	2.141
32768000	7.539×10^9	2.092
65536000	1.589×10^{10}	2.107