Übungszettel 7

Robin Heinemann

January 27, 2017

Aufgabe 7.2

a) Der optimale Fall ist, wenn das Array bereits sortiert ist. Dann wird von der inneren Schleife immer nur der erste Durchlauf ausgeführt. Die äußere Schleife wird genau n mal ausgeführt, damit erhält man $f_1(n) = n$, dies lässt sich zu $n \in \Omega(n) \implies g_1(n) = n$ umformen. Hier muss \$Ω\$- Notation verwendet werde, da es sich um den günstigsten Fall handelt, es gibt Fälle, bei denen $O(n)$ nicht die obere Schranke ist.
b) Der schlechteste Fall tritt auf, wenn das Array genau invers sortiert ist. Die innere Schleife wird also immer genau $n - i$ mal durchlaufen wird, wobei i die Anzahl der Durchläufe der äußeren Schleife bezeichnet. Damit erhält man:

$$f_2(n) = \sum_{i=1}^{n} n - i = \frac{1}{2}(n^2 - n)$$

Dies lässt sich zu:

$$f_2(n) \in O(n^2) \implies g_2(n) = n^2$$

vereinfachen.
c) Analog zum schlechtesten Fall erhält man dann für die innere Schleife $\frac{n-i}{2}$ Durchläufe:

$$f_2(n) = \sum_{i=1}^{n} \frac{n-i}{2} = \frac{1}{4}(n^2 - n)$$

und damit:

$$f_3(n) \in O(n^2) \implies g_3(n) = n^2$$

d)

· ohne Optimierungen insertion sort best case:

n	Zeit in Sekunden	Zeit / $n \log n$
1000000	0.0384659	$3.846\,59 \times 10^{-8}$
1200000	0.0470058	$3.917\,15 \times 10^{-8}$
1440000	0.0567039	$3.937\,77 \times 10^{-8}$
1727999	0.0673467	$3.897\,38 \times 10^{-8}$
2073600	0.0811057	$3.911\,35 \times 10^{-8}$
2488319	0.098592	$3.962\,19 \times 10^{-8}$
2985983	0.116828	$3.912\,55 \times 10^{-8}$
3583180	0.140203	3.9128×10^{-8}
4299816	0.16865	$3.922\,27 \times 10^{-8}$
5159780	0.203377	$3.941\,58 \times 10^{-8}$
6191736	0.243178	$3.927\,46 \times 10^{-8}$
7430083	0.292753	$3.940\,11 \times 10^{-8}$
8916100	0.350108	3.9267×10^{-8}
10699320	0.414265	$3.871\,89 \times 10^{-8}$
12839184	0.50081	$3.900\,63 \times 10^{-8}$
15407021	0.59793	$3.880\,89 \times 10^{-8}$
18488425	0.724838	3.9205×10^{-8}
22186111	0.880652	$3.969\,38 \times 10^{-8}$
26623333	1.04973	$3.942\,91 \times 10^{-8}$
31947999	1.26629	$3.963\,61 \times 10^{-8}$
38337599	1.50603	$3.928\,33 \times 10^{-8}$
46005119	1.83551	3.9898×10^{-8}
55206143	2.18521	$3.958\,28 \times 10^{-8}$
66247372	2.5922	3.9129×10^{-8}
79496847	3.12522}	$3.931\,26 \times 10^{-8}$

Durchschnitt: $3.925\,05 \times 10^{-8}$

insertion sort worst case:

n	Zeit in Sekunden	Zeit / $n \log n$
2000	0.0563335	$1.408\,34 \times 10^{-8}$
2140	0.0637945	$1.393\,02 \times 10^{-8}$
2289	0.0708066	1.3514×10^{-8}
2450	0.0824421	$1.373\,46 \times 10^{-8}$
2621	0.0940372	$1.368\,88 \times 10^{-8}$
2805	0.107795	$1.370\,04 \times 10^{-8}$
3001	0.127077	$1.411\,03 \times 10^{-8}$
3211	0.14668	$1.422\,62 \times 10^{-8}$
3436	0.161834	$1.370\,76 \times 10^{-8}$
3676	0.18396	$1.361\,35 \times 10^{-8}$
3934	0.21371	$1.380\,88 \times 10^{-8}$
4209	0.2442	$1.378\,44 \times 10^{-8}$
4504	0.291699	$1.437\,93 \times 10^{-8}$
4819	0.324696	$1.398\,18 \times 10^{-8}$
5157	0.377055	$1.417\,79 \times 10^{-8}$
5518	0.429574	$1.410\,83 \times 10^{-8}$
5904	0.494398	$1.418\,35 \times 10^{-8}$
6317	0.575462	1.4421×10^{-8}
6759	0.637805	$1.396\,12 \times 10^{-8}$
7233	0.757041	$1.447\,05 \times 10^{-8}$
7739	0.850441	$1.419\,95 \times 10^{-8}$
8281	1.0016	$1.460\,59 \times 10^{-8}$
8860	1.10167	$1.403\,41 \times 10^{-8}$
9481	1.27307	$1.416\,27 \times 10^{-8}$
10144	1.4453}	$1.404\,56 \times 10^{-8}$

Durchschnitt: $1.402\,53 \times 10^{-8}$

insertion sort typical case:

n	Zeit in Sekunden	Zeit / $n \log n$
5000	0.179809	$7.192\,36 \times 10^{-9}$
5350	0.202922	$7.089\,59 \times 10^{-9}$
5724	0.236923	$7.231\,16 \times 10^{-9}$
6125	0.270223	$7.202\,95 \times 10^{-9}$
6553	0.311509	$7.254\,21 \times 10^{-9}$
7012	0.35963	$7.314\,28 \times 10^{-9}$
7503	0.427693	$7.597\,35 \times 10^{-9}$
8028	0.475772	$7.382\,17 \times 10^{-9}$
8590	0.558334	$7.566\,72 \times 10^{-9}$
9192	0.648983	$7.680\,92 \times 10^{-9}$
9835	0.750438	$7.758\,29 \times 10^{-9}$
10524	0.812448	$7.335\,57 \times 10^{-9}$
11260	0.912265	$7.195\,23 \times 10^{-9}$
12049	1.10621	$7.619\,65 \times 10^{-9}$
12892	1.2469	$7.502\,25 \times 10^{-9}$
13795	1.41741	$7.448\,23 \times 10^{-9}$
14760	1.5875	$7.286\,85 \times 10^{-9}$
15794	1.86811	7.4889×10^{-9}
16899	2.0874	$7.309\,42 \times 10^{-9}$
18082	2.37665	$7.268\,97 \times 10^{-9}$
19348	2.74368	$7.329\,27 \times 10^{-9}$
20702	3.09305	7.2171×10^{-9}
22152	3.58881	7.3135×10^{-9}
23702	4.0872	$7.275\,38 \times 10^{-9}$
25361	4.64648	$7.224\,23 \times 10^{-9}$

Durchschnitt: $7.363\,38 \times 10^{-9}$

std::sort:

n	Zeit in Sekunden	Zeit / $n \log n$
100000	0.0448357	3.89438×10^{-8}
107000	0.0485742	3.92005×10^{-8}
114490	0.0526924	3.95112×10^{-8}
122504	0.0572872	3.99146×10^{-8}
131079	0.0620617	4.01804×10^{-8}
140255	0.0672582	4.04635×10^{-8}
150073	0.0706144	3.9478×10^{-8}
160578	0.0765097	3.975×10^{-8}
171818	0.0809247	3.90728×10^{-8}
183845	0.087469	3.92494×10^{-8}
196715	0.0945126	3.94154×10^{-8}
210485	0.0999723	3.87497×10^{-8}
225219	0.10864	3.91384×10^{-8}
240984	0.118354	3.96311×10^{-8}
257853	0.126013	3.92211×10^{-8}
275903	0.136135	3.93856×10^{-8}
295216	0.151025	4.06159×10^{-8}
315881	0.158583	3.96454×10^{-8}
337993	0.168726	3.92121×10^{-8}
361652	0.180095	3.89093×10^{-8}
386968	0.194031	3.89717×10^{-8}
414056	0.207441	3.87356×10^{-8}
443040	0.226052	3.92442×10^{-8}
474052	0.241299	3.8948×10^{-8}
507236	0.258856	3.88472×10^{-8}

Durchschnitt: 3.93774×10^{-8}

· mit Optimierungen insertion sort best case:

n	Zeit in Sekunden	Zeit / n
1000000	0.00180217	1.80217×10^{-9}
1200000	0.00221753	1.84794×10^{-9}
1440000	0.00302139	2.09819×10^{-9}
1727999	0.00383213	2.21767×10^{-9}
2073600	0.00465811	2.24639×10^{-9}
2488319	0.00552709	2.22121×10^{-9}
2985983	0.00680263	2.27819×10^{-9}
3583180	0.00791746	2.20962×10^{-9}
4299816	0.00957455	2.22673×10^{-9}
5159780	0.0114418	2.2175×10^{-9}
6191736	0.013643	2.20343×10^{-9}
7430083	0.0169643	2.28319×10^{-9}
8916100	0.0204979	2.29897×10^{-9}
10699320	0.0237447	2.21927×10^{-9}
12839184	0.028712	2.23628×10^{-9}
15407021	0.0342781	2.22484×10^{-9}
18488425	0.0410346	2.21947×10^{-9}
22186111	0.0506048	2.28092×10^{-9}
26623333	0.0595367	2.23626×10^{-9}
31947999	0.0714738	2.23719×10^{-9}
38337599	0.0865582	2.25779×10^{-9}
46005119	0.103315	2.24572×10^{-9}
55206143	0.132733	2.40431×10^{-9}
66247372	0.171755	2.59263×10^{-9}
79496847	0.195373	2.45763×10^{-9}

Durchschnitt: 2.23054×10^{-9}

insertion sort worst case:

n	Zeit in Sekunden	Zeit / n^2
2000	0.00131307	$3.282\,68 \times 10^{-10}$
2140	0.00151938	$3.317\,71 \times 10^{-10}$
2289	0.00175493	3.3494×10^{-10}
2450	0.00203224	$3.385\,66 \times 10^{-10}$
2621	0.00289405	4.2128×10^{-10}
2805	0.00272793	$3.467\,11 \times 10^{-10}$
3001	0.00347161	$3.854\,77 \times 10^{-10}$
3211	0.00443669	$4.303\,07 \times 10^{-10}$
3436	0.00420283	$3.559\,88 \times 10^{-10}$
3676	0.00480767	$3.557\,82 \times 10^{-10}$
3934	0.00556181	$3.593\,75 \times 10^{-10}$
4209	0.00692586	$3.909\,45 \times 10^{-10}$
4504	0.00730396	$3.600\,49 \times 10^{-10}$
4819	0.00962725	$4.145\,61 \times 10^{-10}$
5157	0.00958699	$3.604\,86 \times 10^{-10}$
5518	0.0116548	$3.827\,74 \times 10^{-10}$
5904	0.0129458	$3.713\,94 \times 10^{-10}$
6317	0.0144537	$3.622\,09 \times 10^{-10}$
6759	0.0172369	$3.773\,07 \times 10^{-10}$
7233	0.0190871	3.6484×10^{-10}
7739	0.0219383	$3.662\,97 \times 10^{-10}$
8281	0.02532	3.6923×10^{-10}
8860	0.0298174	$3.798\,42 \times 10^{-10}$
9481	0.0346395	$3.853\,57 \times 10^{-10}$
10144	0.0427375}	$4.153\,27 \times 10^{-10}$

Durchschnitt: $3.715\,63 \times 10^{-10}$

insertion sort worst case:

n	Zeit in Sekunden	Zeit / n^2
5000	0.00493283	1.97313×10^{-10}
5350	0.00571745	1.99754×10^{-10}
5724	0.00632268	1.92975×10^{-10}
6125	0.0076292	2.03361×10^{-10}
6553	0.00888076	2.06809×10^{-10}
7012	0.00992745	2.01908×10^{-10}
7503	0.0112757	2.00297×10^{-10}
8028	0.0131061	2.03357×10^{-10}
8590	0.0155581	2.10848×10^{-10}
9192	0.0172611	2.04291×10^{-10}
9835	0.020206	2.08897×10^{-10}
10524	0.0228511	2.06322×10^{-10}
11260	0.026403	2.08246×10^{-10}
12049	0.029942	2.06243×10^{-10}
12892	0.0342058	2.05807×10^{-10}
13795	0.0388196	2.03989×10^{-10}
14760	0.0452196	2.07565×10^{-10}
15794	0.0519203	2.08139×10^{-10}
16899	0.0591509	2.07128×10^{-10}
18082	0.0680828	2.0823×10^{-10}
19348	0.07862	2.1002×10^{-10}
20702	0.0894472	2.08709×10^{-10}
22152	0.102976	2.09851×10^{-10}
23702	0.117309	2.08814×10^{-10}
25361	0.13398	2.08309×10^{-10}

Durchschnitt: 2.05487×10^{-10}

std::sort typical case:

n	Zeit in Sekunden	Zeit / $n \log n$
100000	0.00763783	$6.634\,14 \times 10^{-9}$
107000	0.00816836	$6.592\,05 \times 10^{-9}$
114490	0.00882383	$6.616\,52 \times 10^{-9}$
122504	0.00930791	$6.485\,25 \times 10^{-9}$
131079	0.0102184	$6.615\,67 \times 10^{-9}$
140255	0.0109099	$6.563\,59 \times 10^{-9}$
150073	0.0118967	$6.651\,05 \times 10^{-9}$
160578	0.0125743	6.5329×10^{-9}
171818	0.0138146	$6.670\,08 \times 10^{-9}$
183845	0.0147697	$6.627\,53 \times 10^{-9}$
196715	0.0159018	$6.631\,69 \times 10^{-9}$
210485	0.0170402	$6.604\,85 \times 10^{-9}$
225219	0.0181737	$6.547\,22 \times 10^{-9}$
240984	0.0194615	$6.516\,74 \times 10^{-9}$
257853	0.0210726	$6.558\,77 \times 10^{-9}$
275903	0.0227183	$6.572\,72 \times 10^{-9}$
295216	0.0244938	$6.587\,22 \times 10^{-9}$
315881	0.0263135	$6.578\,31 \times 10^{-9}$
337993	0.028485	$6.619\,93 \times 10^{-9}$
361652	0.030154	$6.514\,74 \times 10^{-9}$
386968	0.0329184	$6.611\,75 \times 10^{-9}$
414056	0.0351035	$6.554\,92 \times 10^{-9}$
443040	0.03794	$6.586\,64 \times 10^{-9}$
474052	0.0404175	$6.523\,78 \times 10^{-9}$
507236	0.0432129	$6.485\,08 \times 10^{-9}$

Durchschnitt: $6.579\,33 \times 10^{-9}$

- Analyse:
  - Es zeigt sich wie erwartet für insertion sort worst / typical case ein quadratisches Verhalten und für best case lineares Verhalten. Außerdem bestätigt sich die Annahme, das im typischen Fall nur die Hälfte der Inneren Schleife durchschritten wird.
  - Die Optimierung wirkt sich sehr positiv auf die Laufzeit aus (bis zu 40x, beim typischen Fall), allerdings unterschiedlich stark auf die insertion sort und std::sort
  - Um zu Berechnen, bis zu welchen n Insertion Sort mit std::sort mithalten kann muss man einfach das jeweilige c einsetzen:

$$2.054\,87 \times 10^{-10} n^2 = 6.579\,33 \times 10^{-9} n \log n \implies n \approx 163$$

Aufgabe 7.3

a)

$$\begin{aligned}
t_{64} &= c \cdot f(64) \\
t_{32} &= c \cdot f(32) = 5\,\mathrm{s} \\
c &= \frac{5\,\mathrm{s}}{f(32)} \\
t_{64} &= \frac{f(64)5\,\mathrm{s}}{f(32)}
\end{aligned}$$

Damit erhält man für die verschiedenen Komplexitäten:

$$\begin{aligned}
t_{64} &= \frac{\log_2(64) \cdot 5\,\mathrm{s}}{\log_2(32)} = 6\,\mathrm{s} \\
t_{64} &= \frac{64 \cdot 5\,\mathrm{s}}{32} = 10\,\mathrm{s} \\
t_{64} &= \frac{64\log_2(64) \cdot 5\,\mathrm{s}}{32\log_2(32)} = 12\,\mathrm{s} \\
t_{64} &= \frac{64^2 \cdot 5\,\mathrm{s}}{32^2} = 20\,\mathrm{s} \\
t_{64} &= \frac{2^{64} \cdot 5\,\mathrm{s}}{2^{32}} = 21\,474\,836\,480\,\mathrm{s}
\end{aligned}$$

b)

$$\begin{aligned}
\log_a(n) &= \frac{\log_b(n)}{\log_b(a)} \\
\implies \log_b(a)\log_a(n) &\le \log_b(n)\,\forall\, n \\
\implies \exists\, C : C\log_a(n) &\le \log_b(n)\,\forall\, n, C = \log_b(a)
\end{aligned}$$

c)

Die Reihenfolge ist:

$$\log(n), \sqrt(n), n\log(n), n^2, 2^n$$

$$\lim_{n\to\infty} \frac{\sqrt{n}}{\log n} = \lim_{n\to\infty} \frac{n}{2\sqrt{n}} = \lim_{n\to\infty} \sqrt{n} = \sqrt{\lim_{n\to\infty} n} = \sqrt{\infty} = \infty$$

$$\log(x) = x\log(x) \implies x = 1 \implies \log(x) > 0 \implies x\log(x) > \log(x) \iff x > 1$$

$$\lim_{n\to\infty} \frac{n^2}{n\log(n)} = \lim_{n\to\infty} \frac{n}{\log(n)} = \lim_{n\to\infty} n = \infty$$

$$\lim_{n\to\infty} \frac{2^n}{n^2} = \lim_{n\to\infty} \frac{2n}{\log(n)+1} = \lim_{n\to\infty} 2n = \infty$$