

## Algorithmen und Datenstrukturen (IB) Praktikum - Aufgabenblatt 5 – Einfaches Sortieren

Laufzeit unsortiert:

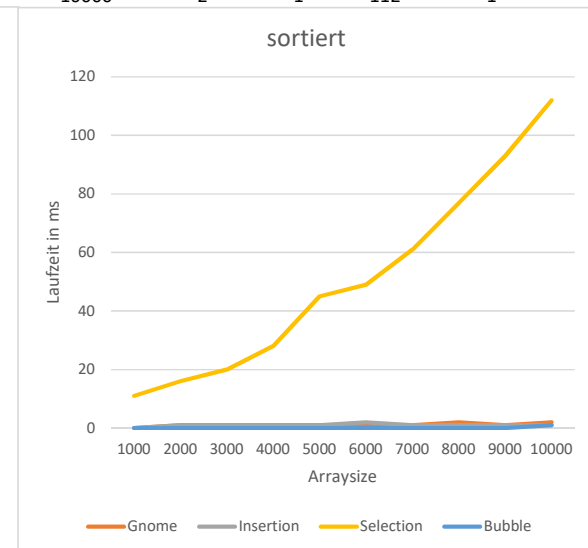
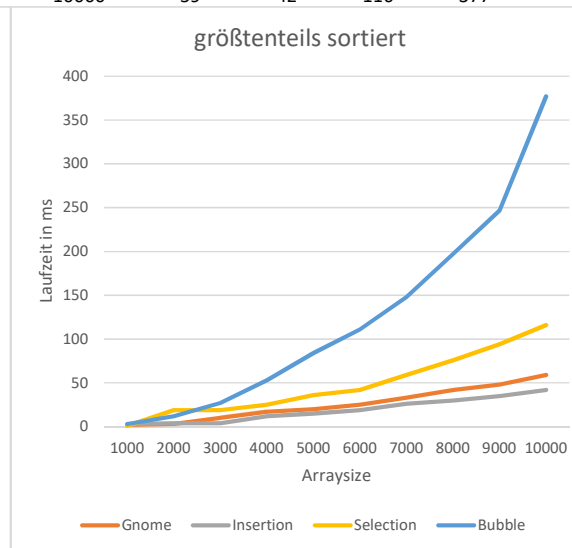
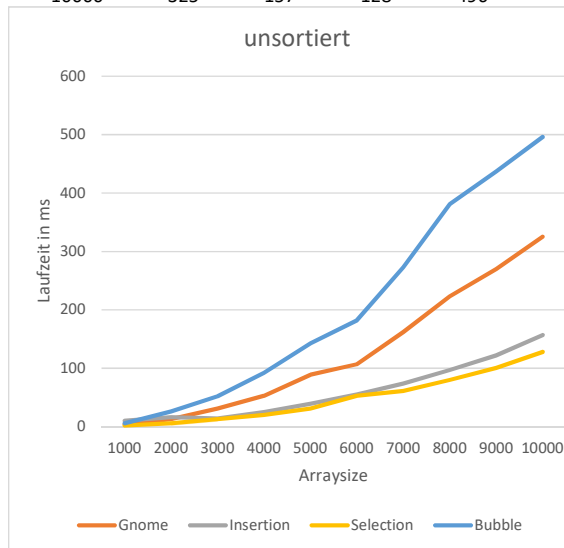
Arraysize	Gnome	Insertion	Selection	Bubble
1000	4	10	2	6
2000	13	16	6	26
3000	31	14	13	52
4000	53	25	20	92
5000	89	39	31	143
6000	107	55	53	182
7000	162	74	61	273
8000	223	97	80	381
9000	270	122	101	437
10000	325	157	128	496

Laufzeit größtenteils sortiert

Arraysize	Gnome	Insertion	Selection	Bubble
1000	2	3	1	3
2000	3	4	19	12
3000	10	4	19	27
4000	17	12	25	53
5000	20	15	36	84
6000	25	19	42	111
7000	33	26	59	148
8000	42	30	76	197
9000	48	35	94	247
10000	59	42	116	377

Laufzeit sortiert

Arraysize	Gnome	Insertion	Selection	Bubble
1000	0	0	11	0
2000	1	1	16	0
3000	1	1	20	0
4000	1	1	28	0
5000	1	1	45	0
6000	1	2	49	0
7000	1	1	61	0
8000	2	1	77	0
9000	1	1	93	0
10000	2	1	112	1



Bei unsortierten Eingaben würde ich den Selection- oder Insertion-Algorithmus wählen

Bei größtenteils sortierten Eingaben würde ich den Insertion- oder Gnome-Algorithmus wählen

Bei sortierten Eingaben würde ich den Gnome-Algorithmus wählen

Welcher Algorithmus ist in Situationen vorzuziehen, in denen das Vertauschen teuer ist, z.B. weil es sich um große Datensätze handelt?

Es ist der Selection-Algorithmus zu wählen, da er die wenigsten Vertauschungen benötigt

Wie verhält sich bei den betrachteten Eingaben jeweils die Laufzeit für teilweise unsortierte Eingaben oder Eingaben mit wenigen Vertauschungen zum völlig unsortierten Fall?

Bei einer höheren Sortierung reduziert sich die Laufzeit, außer beim Selection-Algorithmus, dort bleibt sie gleich