

Insertion Sort

Während Selection Sort jeweils alle noch nicht bearbeiteten Elemente betrachtet hat, um das kleinste zu finden, orientiert sich **Insertion Sort** nach links: Es betrachtet jeweils ein Element und rückt dieses dann soweit nach links, bis es an seiner korrekten Position innerhalb der bislang betrachteten Elemente gelandet ist.

Beispiel

Für die Zeichenkette „ZEBRASSINDGELB“ sieht das dann folgendermaßen aus:

		a[]													
i	j	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		Z	E	B	R	A	S	S	I	N	D	G	E	L	B
1	0	E	Z	B	R	A	S	S	I	N	D	G	E	L	B
2	0	B	E	Z	R	A	S	S	I	N	D	G	E	L	B
3	2	B	E	R	Z	A	S	S	I	N	D	G	E	L	B
4	0	A	B	E	R	Z	S	S	I	N	D	G	E	L	B
5	4	A	B	E	R	S	Z	S	I	N	D	G	E	L	B
6	5	A	B	E	R	S	S	Z	I	N	D	G	E	L	B
7	3	A	B	E	I	R	S	S	Z	N	D	G	E	L	B
8	4	A	B	E	N	R	S	S	Z	D	G	E	L	B	
9	2	A	B	D	E	I	N	R	S	S	Z	E	L	B	
10	4	A	B	D	E	G	I	N	R	S	S	Z	E	L	B
11	4	A	B	D	E	E	G	I	N	R	S	S	Z	L	B
12	7	A	B	D	E	E	G	I	L	N	R	S	S	Z	B
13	2	A	B	B	D	E	E	G	I	L	N	R	S	S	Z
		A	B	B	D	E	E	G	I	L	N	R	S	S	Z

Hellgrau sind Elemente die sich im aktuellen Schritt nicht bewegen.

Die roten Elemente wurden im aktuellen Schritt an dieser Stelle "einsortiert".

Die schwarzen Elemente wurden um einen Platz nach rechts bewegt, damit das rote Element eingefügt werden konnte.



(A1)

- Implementiere im Bluej-Projekt <https://codeberg.org/qg-info-unterricht/algs4-sort-bluej> Selectionsort.
- Erzeuge mit Hilfe der draw-Methode eine Veranschaulichung des Sortiervorgangs wie im Bild oben. Du muss dazu die Methode draw kopieren und anpassen, um die Färbung für Insertionsort korrekt zu erzeugen.



(A2)

wenn alle Elemente der zu sortierenden Liste denselben Sortier-Schlüssel haben - welches Verfahren ist dann effizienter. Insertion-Sort oder Selection-Sort?

Tipp

Veranschauliche in beiden Verfahren das sortieren einer Zeichenkette aus gleichen Buchstaben und überlege dir, welches Verfahren effizienter ist.

		a[]									
i	j	0	1	2	3	4	5	6	7	8	9
		a	a	a	a	a	a	a	a	a	a
1	1	a	a	a	a	a	a	a	a	a	a
2	2	a	a	a	a	a	a	a	a	a	a
3	3	a	a	a	a	a	a	a	a	a	a
4	4	a	a	a	a	a	a	a	a	a	a
5	5	a	a	a	a	a	a	a	a	a	a
6	6	a	a	a	a	a	a	a	a	a	a
7	7	a	a	a	a	a	a	a	a	a	a
8	8	a	a	a	a	a	a	a	a	a	a
9	9	a	a	a	a	a	a	a	a	a	a
		a	a	a	a	a	a	a	a	a	a

		a[]									
i	min	0	1	2	3	4	5	6	7	8	9
		a	a	a	a	a	a	a	a	a	a
0	0	a	a	a	a	a	a	a	a	a	a
1	1	a	a	a	a	a	a	a	a	a	a
2	2	a	a	a	a	a	a	a	a	a	a
3	3	a	a	a	a	a	a	a	a	a	a
4	4	a	a	a	a	a	a	a	a	a	a
5	5	a	a	a	a	a	a	a	a	a	a
6	6	a	a	a	a	a	a	a	a	a	a
7	7	a	a	a	a	a	a	a	a	a	a
8	8	a	a	a	a	a	a	a	a	a	a
9	9	a	a	a	a	a	a	a	a	a	a

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:algorithmen:sorting:insertionsort:start?rev=1675887036>

Last update: **08.02.2023 20:10**

