

Aufgabe 10:

Was passiert bei Sortieren durch Einfügen, wenn die gegebene Folge bereits

(a) aufsteigend oder

(b) absteigend sortiert ist?

*(c) Was passiert, wenn alle Elemente gleich sind? Ist es möglich, dass man
bloß $O(n)$ Zeit benötigt?*

Sortieren durch Einfügen (aus der Vorlesung)

Iteration:

Nach dem i -ten Schritt sind die ersten i Elemente sortiert.

Das nächste Element (das $i+1$ -te) wird dann

an er richtigen Stelle zwischen die ersten i Elemente eingefügt.

(a) richtige Stelle finden

(b) einfügen

Speicherung in einem Array:

(a) z.B. durch sequentielle Suche

*(b) alle Elemente rechts von der Einfügetabelle werden um eins nach rechts
verschoben*

Speicherung als verkettete Liste:

(a) ist nur durch sequentielle Suche möglich

(b) Umhängen einiger Zeiger

- Beim i -ten Element benötigt man $i-1$ Vergleiche und keine Verschiebungen, wenn man von rechts nach links sucht, denn man nimmt sich das i -te Element und vergleicht es mit den vorherigen. Da es kein größeres gibt, hat man $i-1$ Vergleiche und weil das Element an derselben Stelle bleibt, gibt es auch keine Verschiebungen. Da die Anzahl von Vergleichen gleich bleibt, ist die Laufzeit $O(n)$. Wenn man von links nach rechts sucht, hat man ebenfalls keine Verschiebungen, aber insgesamt $O(n^2)$ Vergleiche.
- Beim i -ten Element benötigt man 1 Vergleich, wenn man von links nach rechts sucht, und $i-1$ Verschiebungen, denn man nimmt das i -te Element und vergleicht es mit dem ersten. Weil es größer ist wird das i -te Element an erster Stelle eingefügt, das heißt man hat nur ein Vergleich und $i-1$ Verschiebungen. Da die Anzahl von Verschiebungen gleich bleibt, ist die Laufzeit $O(n^2)$. Wenn man von rechts nach links die Einfügestelle sucht, hat man $O(n^2)$ Vergleiche aber trotzdem $O(n^2)$ Verschiebungen, dies bedeutet man braucht auch $O(n^2)$.

Anmerkung: Wenn man die Zahlen in einer Liste anstatt in einem Array hat, so hat man auch hier nur $O(n)$ Verschiebungen.

- Wenn Sortieren durch Einfügen stabil ist, verhält es sich als wenn die Folge wie in a) schon sortiert ist, also in $O(n)$, wenn es nicht stabil ist verhält es sich wie in b), also $O(n^2)$ wenn im Array implementiert oder $O(n)$ wenn in einer Liste implementiert.

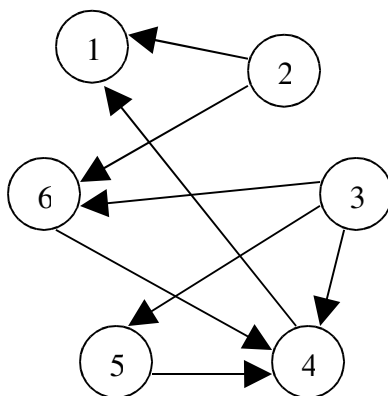
Aufgabe 11:

Beantworten Sie die vorige Frage für Sortieren durch Auswahl.

Man hat die folgende Anzahl an Vergleichen egal wie die Folge sortiert ist. Wenn die Folge aufsteigend sortiert ist bzw. alle Elemente gleich sind findet keine Verschiebung statt. Bei absteigend sortierten Folgen findet allerdings bei jedem Durchgang eine Verschiebung statt, wodurch die Laufzeit auch bei $O(n^2)$ bleibt

Aufgabe 15:

Geben Sie alle möglichen topologischen Sortierungen für folgende Eingabe an: $n = 6$ und $\{(6, 4), (3, 5), (4, 1), (3, 4), (5, 4), (2, 1), (2, 6), (3, 6)\}$.



Sortierung:

2 3 5 6 4 1

2 3 6 5 4 1

3 2 5 6 4 1

3 2 6 5 4 1

3 5 2 6 4 1