InsertionSort (Sortieren durch Einfügen)

InsertionSort verfährt nach folgender Methode:

Nehmen wir an, a[1],...,a[i-1] seien bereits sortiert, d.h. a[1].key ≤ ... ≤ a[i-1].key. Dann wird das i-te Element a[i] an der richtigen Stelle in die Folge a[1],...,a[i] eingefügt. Das Einfügen geschieht so, dass man a[i].key der Reihe nach mit a[i-1].key, a[i-2].key,.. vergleicht und dann das Element a[j] dabei jeweils um eine Position nach rechts verschiebt, für j=i-1,i-2,...,wenn a[j].key > a[i],key ist. Sobald a[j].key ≤ a[i].key, hat man die Stelle gefunden, na der das Element a[i] eingefügt werden kann, nämlich die Position j+1.

Dazu ein Bsp.:

Feld I=2,15,43,17,14,8,47

2	15	43	17	4	8	47

2	15	43	17	4	8	47

2	15	43	17	4	8	47

2	15	17	43	4	8	47

2	4	15	17	43	8	47

2	4	8	15	17	43	47

Analyse:

- WorstCase : $O(n^2)$

  Warum: Zum Einfügen des i. Elementes werden mindestens 1 und maximal i Vergleiche und mindestens zwei oder höchstens i+1 Bewegungen von Datensätzen ausgeführt.

  → $O(n*n-1)=O(n^2)$

- AverageCase: $O(n^2)$
- BestCase (vorsortiert): O(n)

FAZIT:

- InsertionSort profitiert im Gegensatz zu SelectionSort von einer Vorsortierung
- Iterativer Algorithmus
- In-Place Algorithmus
- Vorgehen wie bei Skat Spiel
- Worstcase $O(n^2)$
- AverageCase $O(n^2)$