

Sortieren durch direktes Einfügen

Beschreibung Diese Methode liegt folgende Idee zugrunde:
Das zu sortierende Array wird gedanklich in einen linken und rechten Teil getrennt.
Der linke Teil ist eine bereits sortierte Sequenz, in die die Elemente des rechten Teils nacheinander einsortiert werden müssen.
Zu Beginn des Sortiervorgangs besteht der bereits sortierte linke Teil nur aus dem ersten Element des Arrays (jedes Element stellt für sich allein eine sortierte Sequenz der Länge eins dar).
Alle anderen Elemente gehören zum unsortierten rechten Teil.
Solange nun der rechte Teil nicht leer ist, wird das jeweils erste Element des rechten Teils in die sortierte Sequenz eingefügt.
Einfügen heißt dabei, das Element solange mit seinem linken Nachbarn zu vertauschen, bis dieser entweder kleiner ist oder der linke Rand des Arrays erreicht wird (dieses Element ist dann das kleinste bisher gefundene).

Beispielfunktion

```
void InsertSort(int *Array, int Anzahl)
{
    int i; // erstes Element im unsortierten Teil
    int j; // Laufindex zum Einsortieren
    int temp; // fuer Austausch zweier Elemente

    for (i = 1; i < Anzahl; i++)
    {
        j = i;
        while (j > 0 && *(Array + j) < *(Array + j - 1))
        {
            temp = *(Array + j);
            *(Array + j) = *(Array + j - 1);
            *(Array + j - 1) = temp;
            j--;
        }
    }
}
```

Sortieren durch direktes Einfügen

| | | | | | | | | |
|----|--|----|--|----|--|----|--|----|
| 13 | | 6 | | 27 | | 9 | | 1 |
| 6 | | 13 | | 27 | | 9 | | 1 |
| 6 | | 13 | | 27 | | 9 | | 1 |
| 6 | | 9 | | 13 | | 27 | | 1 |
| 1 | | 6 | | 9 | | 13 | | 27 |

