

- **Ausgangspunkt:** Ein Feld von Elementen
- **Sortieren des Feldes** erfordert den Vergleich von jeweils zwei Elementen
- **Funktionszeiger für die Vergleichsfunktion:**
Sortieralgorithmus nur einmal implementieren, Vergleich dynamisch austauschbar

```
typedef int Element;  
typedef struct {  
    Element *elems;  
    unsigned int size;  
} Field;
```

```
Field sort(Field f) {  
    ...  
    Element l, r;  
    if( l > r ) {  
        ...  
    }  
    ...  
}
```

```
typedef int      (*CompareOp)(Element, Element);  
  
Field sort(Field f, CompareOp cmp) {  
    ...  
    Element l, r;  
    if( cmp(l, r) > 0 ) {  
        ...  
    }  
    ...  
}
```

- **Anwendungsbeispiel:** Feld ganzer Zahlen, aufsteigende Sortierung

```
int icmp(int l, int r) {  
    return l - r;  
}
```

```
Field fld = ...;  
fld = sort(fld, icmp);
```