

Einfacher Quick Sort in C++

Frank Birbacher

Inform GmbH

13. Februar 2014

QuickSort

3 1 4 2

QuickSort

1 2 3 4

Teile-Und-Herrsche

- Pivot auswählen
- partitionieren
- rekursiv aufrufen

Signatur

```
template<typename lter>
void quick_sort(
    lter first,
    lter last
)
{
```

Pivot auswählen – I

```
lter const pivot = first++;
```

Pivot auswählen – II

```
if(first == last) return;  
lter const pivot = first++;
```

Partitionsbedingung – I

```
typedef typename std::iterator_traits<Iter>  
    ::reference reference;
```

```
auto const lessThanPivot  
    = [=](reference current)  
    { return current < *pivot; };
```


Partitionsbedingung – II

Boost:

```
using boost::lambda::_1;
```

```
auto const lessThanPivot  
    = _1 < *pivot;
```

Partitionieren

```
Iter const middle =  
    std::partition(first, last, lessThanPivot);
```

Rekursiv aufrufen

```
quick_sort(first, middle);  
quick_sort(middle, last);
```

Pivot einfügen

```
std::rotate(pivot, first, middle);
```

Zusammen

```
template<typename Iter>
void quick_sort(Iter first, Iter last) {
    if(first == last) return;
    Iter const pivot = first++;

    auto const lessThanPivot = _1 < *pivot;
    Iter const middle = std::partition(first, last, lessThanPivot);

    quick_sort(first, middle);
    quick_sort(middle, last);

    std::rotate(pivot, first, middle);
}
```