Counting sort

```
clear array count[0..k];

for i:=1 to n
  count[A[i].key]++

pos[0]:=1;
for i:=1 to k
  pos[i]:=pos[i-1]+count[i-1];
// now pos[i] is the first position where
// integer i will come in the sorted array B

for i:=1 to n
  B[pos[A[i].key]]:=A[i];
  pos[A[i].key]++;
```

Radix sort

```
for i:=1 to d
  use counting sort to sort A[1..n] by
  the d-th least significant digit (i.e. k=10)
  // inv: array is sorted by last i digits
```

Sort	Running time	Analysis
Min Sort	$\Theta(n^2)$	worst-case
Merge Sort	$\Theta(n \log n)$	worst-case
Heap Sort	$\Theta(n \log n)$	worst-case
Quick Sort	$\Theta(n^2)$	worst-case
(randomized)	$\Theta(n \log n)$	expected
Counting Sort numbers from $[0, k]$	$\Theta(n + k)$	worst-case
Radix Sort d digit numbers	$\Theta(dn)$	worst-case