Sorting

- Organizing data into
  - ascending oder *or*
  - descending order
- Using algorithms

Categories

- **Internal Sorting**
  - **data fit in memory**

- **External Sorting**
  - **data do not fit in memory at once**

Internal Sorting

- Simplifications:
  - Array of elements
  - Only integers

- Algorithms:
  - Easy: O(N^2)
  - Shell Sort: o(N^2)
  - More complicated: O(N*log(N))

- General-purpose sorting algorithm:
  - N*log(N) comparisons

Preliminaries

- Preconditions:
  - A - array
  - N - size of the array

- Postconditions:
  - A is sorted in ascending order
  - N is unchanged

Selection Sort

```
SelectionSort(int A[], int N)
{
 for (int Last = N-1; Last >=1;
     --Last)
  {
   int L = Largest (A, Last+1);
   Swap (A[L], A[Last]);
  }
}
```

10	20	31	5	12

10	20	12	5	31

10	5	12	20	31

10	5	12	20	31

5	10	12	20	31

\pn

Selection Sort (2)

```
SelectionSort(int A[], int N)
{
 for (int Last = N-1; Last >=1; --Last)    <- 1 + N + N-1= 2*N
  {
   int L = Largest (A, Last+1);            <- (Last+1) - 1 = Last
   Swap (A[L], A[Last]);                   <- 3*(N-1)
  }
}

O(N^2) - comparisons
O(N)    - moves
-----------
 O(N^2) - operations
```

Bubble Sort

```
BubbleSort (int A[], int N)
{
 bool Sorted = false;

 for (int Pass = 1;
     (Pass < N) && !Sorted; ++Pass)
  {
   Sorted = true;
   for (int Index = 0;
       Index < N-Pass; ++Index)
    if (A[Index] > A[Index+1])
     {
      Swap (A[Index], A[Index+1]);
      Sorted = false;
     }
  }
}
```

10	20	31	5	12

10	20	31	5	12

10	20	5	31	12

10	20	5	12	31

\pn

Bubble Sort (2)

- Each step requires:
  - N - i exchanges
  - N - i comparisons
- ----------------
- Worst case: O(N^2)
- Best case: O(N) (comparisons)

Insertion Sort

```
InsertionSort (int A[], int N)
{
 for (int Un = 1; Un < N; ++Un)
  {
   int item = A[Un];
   for (int Loc = Un;
       (Loc > 0)&& (A[Loc-1] > item);
       --Loc)
     A[Loc] = A[Loc-1];
    A[Loc] = item;
  }
}
```

10	20	31	5	12

10	20	31	5	12

10	20	31	5	12

5	10	20	31	12

5	10	12	20	31

\pn

Insertion Sort (2)

- **Worst Case:**
  - **O(N^2)**
- **Best Case:**
  - **O(N)**