Sorting algorithms I.

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Sorting algorithms

- puts elements of a list in a certain order (alphabetical, numbered)
- Pair key-value – sorting according to key, value is not taken into account
- Classification
  - Stable vs. unstable – keeps order of values with the same key
  - Type of sorting
    - selection
    - insertion
    - exchanging
    - merging

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Sorting algorithms

- Simple algorithms
  - Buble sort
  - Heap sort
  - Insertion sort
  - Merge sort
  - Quicksort
  - Selection sort
- Algorithms based on other principle
  - Bucket sort
  - Radix sort
  - Counting sort

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Bubble sort

- Simple to implement
- Universal, local (in-place, no need of extra memory)
- The algorithm starts at the beginning of the data set. It compares the first two elements, and if the first is greater than the second, it swaps them. It continues doing this for each pair of adjacent elements to the end of the data set. It then starts again with the first two elements, repeating until no swaps have occurred on the last pass.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Bubble sort

```
procedure bubbleSort( A : list of sortable items )
    n = length(A)
    repeat
        swapped = false
        for i = 1 to n-1 inclusive do
                if A[i-1] > A[i] then
                        swap( A[i-1], A[i] )
            swapped = true
          end if
        end for
    until not swapped
end procedure
```

Heap sort

- a comparison-based sorting algorithm
- Not stable
- Using data structure heap and its properties

Heap sort

```
procedure heapsort(a, count) is
    input: an unordered array a of length count
    heapify(a, count)
   end ← count - 1
    while end > 0 do
        swap(a[end], a[0])
       (the heap size is reduced by one)
       end ← end - 1
       (the swap ruined the heap property, so restore it)
       siftDown(a, 0, end)
```

Insertion sort

- A simple sorting algorithm that builds the final sorted array (or list) one item at a time
- Simple implementation
- Efficient for (quite) small data sets
- Efficient for data sets that are already substantially sorted
- Stable, on-line, in-place

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Insertion sort

```
for i = 1 to length(A)
    j ← i
    while j > 0 and A[j-1] > A[j]
        swap A[j] and A[j-1]
        j ← j - 1
    end while
end for
```