Sorting algorithms II.

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

Merge sort

- is an efficient, general-purpose, comparison-based sorting algorithm
- Stable, divide and conquer algorithm, easy to paralelized
- Worst and average time: *O(N*log*N)*
- Extra memory needs: array of size *N*

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

Merge sort

- Algoritmus

```
mergesort(m)
        var list left, right
        if length(m) ≤ 1
                return m
        else
                middle = length(m) / 2
                for each x in m up to middle
                        add x to left
                for each x in m after middle
                        add x to right
                left = mergesort(left)
                right = mergesort(right)
                result = merge(left, right)
                return result
```

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

Quicksort

- an efficient sorting algorithm
- It takes: $O(N\log N) – O(N^2)$
- Divide and conquer algorithm, not stable, in-place
- Recursive
- The steps are:
  - Pick an element, called a pivot, from the array.
  - Partitioning: reorder the array so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it (equal values can go either way). After this partitioning, the pivot is in its final position. This is called the partition operation.
  - Recursively apply the above steps to the sub-array of elements with smaller values and separately to the sub-array of elements with greater values.

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

Quicksort

- Pivot selection – ideal is median
  - First element (or any fix position) –
  - Random element –
  - Median of three (five…) – or other number of elements from fix or random positions
- Pivot selection affects sorting
- In average Quicksort is the fastet known universal algorithm for sorting of arrays in computer memory

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

Selection sort

- Simple algorithm
- Time complexity $O(N^2)$
- Efficient for small sets
- Universal, local, not stable
- The algorithm divides the input list into two parts: the sublist of items already sorted, which is built up from left to right at the front (left) of the list, and the sublist of items remaining to be sorted that occupy the rest of the list.
- Initially, the sorted sublist is empty and the unsorted sublist is the entire input list.
- The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist, exchanging (swapping) it with the leftmost unsorted element (putting it in sorted order), and moving the sublist boundaries one element to the right

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

Comparison of sorting algorithms

Name	Time complexity			Extra memory	Stable	Natural	Method
	Minimum	Average	Maximum				
Bubble sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	yes	yes	Exchange
Heapsort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$	no	no	Heap, exchange
Insertion sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	yes	yes	Inserting
Merge sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(\log n)$	yes	yes	Merging
Quicksort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$	no	no	Exchanging
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	no	no	Selection

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

Zdroj: www.wikipedia.org

Bucket sort

- distributing the elements of an array into a number of buckets
- Časová náročnost: *O(n*k)*, kde *k=n/m,* vstupní data *n,* počet přihrádek *m*.
- Assumptions:
  - Suitable for evenly distributed input data.
  - The ordering algorithm must be stable
- Bucket sort works as follows:
  - Set up an array of initially empty "buckets".
  - Scatter: Go over the original array, putting each object in its bucket.
  - Sort each non-empty bucket.
  - Gather: Visit the buckets in order and put all elements back into the original array.
- Advantages: Well parallelized, not all data in memory at one time

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

Radix sort

- a non-comparative integer sorting algorithm that sorts data with integer keys by grouping keys by the individual digits which share the same significant position and value
- LSD (Least Significant Digit)
- MSD (Most Significant Digit)
- Time complexity: $O((z+n)*\log_z u)$, kde z je základ zvolené číselné soustavy, n počet čísel na vstupu a u je maximální rozmezí čísel na vstupu
- Not for unlimited inputs
- Example LSD radix: 170, 45, 75, 90, 802, 2, 24, 66 ⇒ 170, 90, 802, 2, 24, 45, 75, 66 ⇒ 802, 2, 24, 45, 66, 170, 75, 90 ⇒2, 24, 45, 66, 75, 90, 170, 802

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

Counting sort

- only suitable for direct use in situations where the variation in keys is not significantly greater than the number of items
- Stable
- Time complexity: *O(N+M)*
- Extra memory: *O(M)*

- Algorithm:

```
for x in input:
    count[key(x)] += 1
total = 0
for i in range(k):   # i = 0, 1, ... k-1
    oldCount = count[i]
    count[i] = total
    total += oldCount
for x in input:
    output[count[key(x)]] = x
    count[key(x)] += 1
return output
```

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