

20. Řadící algoritmy II (Quick - sort)

Jeden z nejrychlejších běžných algoritmů řazení porovnávání prvků. Vymyslel jej Sir **Charles Antony Richard Hoare** v roce 1962.

Quick Sort

Rozdělí řazené posloupnosti čísel na dvě přibližně stejné části.

V jedné části jsou čísla větší a ve druhé menší, než nějaká zvolená hodnota (pivot).

Pokud je tato hodnota zvolena dobře, jsou obě části přibližně stejně velké.

Pokud budou obě části samostatně seřazeny, je seřazené i celé pole.

Obě části se pak rekurzivně řadí stejným postupem (Lze zapsat i iteračně).

Složitost = n^2 (nejhorší možný případ)

```
public static void quick(int[] array, int left, int right) {
    int i = left;
    int j = right;
    int pivot = array[(left+right)/2];
    do{
        //while(array[i] > pivot) i++;
        //while(array[j] < pivot) j--;
        while(array[i] < pivot) i++;
        while(array[j] > pivot) j--;
        if(i <= j){
            int tmp = array[i];
            array[i] = array[j];
            array[j] = tmp;
            i++;
            j--;
        }
    }while(i<j);
    if((j-left) > 0){
        quick(array, left, j);
    }
    if((right-i) > 0){
        quick(array, i, right);
    }
}
```

Lomuto Quick Sort

Nico Lomuto v roce 1984 přišel s vylepšením Quick Sortu. Principem je postupné umísťování prvků menších jak pivot doleva. Levá půlka pole se zvětšuje. Výhodou je méně porovnání, ale nevýhodou je více přesunů než Hoareho metoda.

Složitost závisí na zvolení vhodného pivotu, může degradovat na N^2 .