

# ITY – 5. projekt

QuickSort

---

Mušková Kateřina

15. listopadu 2019

# Definice problému

---

## Definice problému

Na vstupu je posloupnost prvků  $S = (S_1, S_2, \dots, S_N)$ .

Na výstupu očekáváme posloupnost  $S = (S'_1, S'_2, \dots, S'_N)$ , pro kterou platí:

- Tato posloupnost je seřazená:

$$S'_1 \leq S'_2 \leq \dots \leq S'_N$$

- Obsahuje stejná data (je permutací původní posloupnosti). [1]

## Quick sort

---

QuickSort v roce 1962 poprvé popsal sir Charles Antony Richard Hoare.

Jde o je velmi rychlý nestabilní algoritmus, fungující na principu rozděl a panuj.

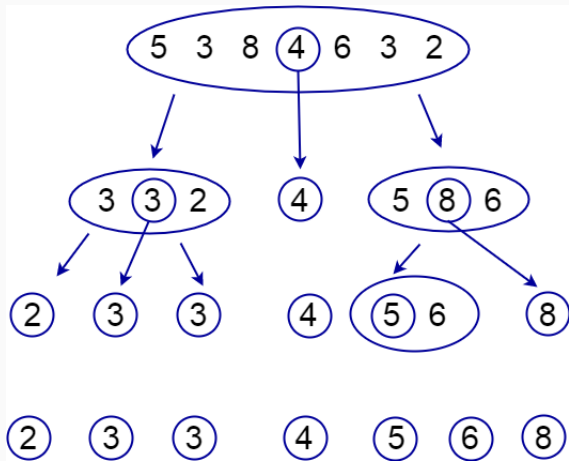
Chová se dobře jak na malých, tak na velkých polích a je paměťově nenáročný.[2]

Jeden prvek seznamu se vybere jako tzv. *pivot*.

Následně se prvky seskládají tak, aby nalevo byly všechny hodnoty menší a napravo větší pivotu.

Pivot jde samozřejmě také možné prohazovat.

Totéž se pak opakuje v obou skupinkách vlevo a vpravo. [3]



Samotný kód se pak skládá ze dvou částí:

- Hlavní část, která má na starosti dělení na 2 části.
- Partition pak přehazuje prvky v závislosti na pivotu.



```
quicksort(A, lo, hi) is
    if lo < hi then
        p := partition(A, lo, hi)
        quicksort(A, lo, p - 1)
        quicksort(A, p + 1, hi)
```

```
partition(A, lo, hi) is
    pivot := A[hi]
    i := lo
    for j := lo to hi - 1 do
        if A[j] < pivot then
            swap A[i] with A[j]
            i := i + 1
    swap A[i] with A[hi]
    return i
```

Výkonnost quicksortu je dána volbou pivota.

Pokud jej volíme ideálně, tak dojde při každém rekurzivním volání k rozpůlení pole a vystačíme si tedy s  $\log_2 n$  voláními, v nichž popřehazujeme až  $n$  prvků. Složitost tohoto případu je proto  $(n \cdot \log_2 n)$ .

Na druhou stranu, pokud nemáme štěstí a pivota volíme špatně, může se časová náročnost vyšplhat až k  $n^2$  [3].

## Volba pivota

- První prvek – popřípadě kterákoli jiná fixní pozice - velice nevýhodná pro byť jen částečně seřazené posloupnosti.
- Náhodný prvek – často používaná metoda.
- Median – ideální případ. V praxi to ovšem vždy neplatí, protože výpočet mediánu může být při řazení různých struktur různě náročný a celý algoritmus se jeho výpočtem může zpomalovat.
- Metoda mediánu tří – případně pěti či sedmi, ... Pomocí pseudonáhodného algoritmu (používají se i fixní pozice, typicky první, prostřední a poslední) se vybere několik prvků z řazené posloupnosti, ze kterých se použitím některého primitivního řadicího algoritmu najde medián a ten je zvolen za pivot.[4]

## Použitá literatura



*Řadicí algoritmus* [online].

**Poslední zmena 30. 10. 2018 [cit. 1. května 2019].**

Dostupné na: <https://cs.wikipedia.org/wiki/>.



*Quick sort (rychlé řazení)* [online].

**[cit. 1. května 2019].**

Dostupné na: <http://voho.eu/wiki/quick-sort/>.



*Quicksort* [online].

**[cit. 1. května 2019].**

Dostupné na:

<https://www.algoritmy.net/article/10/Quicksort>.



*Quicksort* [online].

**Poslední zmena 27. 4. 2018 [cit. 2. dubna 2019].**

Dostupné na:

<https://moodle.fit.cvut.cz/mod/page/view.php?id=60922>