

KIV / PRO

Výpočet mediánu

Zadání

Najděte v dostupné literatuře nebo vymyslete co nejlepší algoritmus pro výpočet mediánu. Nezapomeňte na citaci zdrojů. Kritéria kvality v sestupném pořadí jsou: výpočetní složitost, jednoduchost a implementační nenáročnost, paměťová spotřeba.

Definice

Podle (1) lze medián definovat jako Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Analýza problému

Nalezení mediánu v zadané číselné posloupnosti lze řešit několika více či méně efektivními způsoby, existující algoritmy lze rozdělit zhruba do následujících skupin:

Algoritmy využívající seřazení posloupnosti

Vzhledem k tomu, že medián se v seřazené posloupnosti délky $2n + 1$ nachází na n -tém místě, můžeme posloupnost nejprve seřadit a následně prohlásit prostřední prvek za medián. Pro tento účel se nejlépe hodí algoritmus quicksort - viz (2). Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Funkci tohoto algoritmu můžeme popsat pseudokódem uvedeným jako Algoritmus 1 (převzato z (3)):

```
Function quicksort(array')
  create empty lists 'less' and 'greater'
  if length('array') ≤ 1
    return 'array' // an array of zero or one elements is already sorted
  select and remove a pivot value 'pivot' from 'array'
  for each 'x' in 'array'
    if 'x' ≤ 'pivot' then append 'x' to 'less'
    else append 'x' to 'greater'
  return concatenate(quicksort('less'), 'pivot', quicksort('greater'))
```

Algoritmus 1: Pseudokód algoritmu quicksort.

Algoritmy fungující bez seřazení posloupnosti

Pro samotné nalezení mediánu ovšem není nezbytně nutné celou posloupnost řadit a ani to není efektivní, neboť seřazení posloupnosti zbytečně navyšuje časovou složitost algoritmu. Místo toho můžeme použít např. algoritmus... Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Jiné typy algoritmů

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Srovnání existujících algoritmů

Výhody a nevýhody jednotlivých algoritmů jsou zřejmé z předchozího textu – algoritmy pracující na základě seřazení posloupnosti potřebují k nalezení algoritmu obvykle $O(n \cdot \log n)$ kroků, což z nich činí nejpomalejší variantu algoritmu pro nalezení algoritmu. Vedlejším produktem jejich činnosti je také seřazení posloupnosti, což nemusí být žádoucí. Naproti tomu, algoritmy typu... Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Závěr

V praxi budeme pro nalezení mediánu používat především algoritmy typu *xyz*, protože využijeme v první řadě tu jejich vlastnost, že *xyz*. Proin sed eleifend turpis. Phasellus quis velit sed lacus consequat dignissim. Suspendisse potenti. Sed aliquet iaculis est id auctor. Morbi nec lacus sem. Aenean pulvinar, ipsum in laoreet scelerisque, elit dui sollicitudin odio, quis molestie neque orci vulputate nisl. Sed eros elit, imperdiet quis posuere in, iaculis nec nunc. Cras a quam ligula, at viverra felis.

Bibliografie

1. **Weisstein, Eric W.** Statistical Median. *MathWorld--A Wolfram Web Resource*. [Online] [Citace: 15. 10 2011.] <http://mathworld.wolfram.com/StatisticalMedian.html>.
2. **Hoare, C. A. R.** Quicksort. *The Computer Journal*. 1962, Sv. 5, 1, stránky 10 - 16.
3. Quicksort. *Wikipedia, The Free Encyclopedia*. [Online] 18. 10 2011. [Citace: 19. 10 2011.] <http://en.wikipedia.org/wiki/Quicksort>.