



EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVYOP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Řazení prvků v poli

Select sort

Projekt
DUMCZ.1.07/1.5.00/34.1009
VY_32_INOVACE_267

Ing. Karel Johanovský

Střední průmyslová škola Jihlava

2013

Identifikační údaje

Projekt	<i>Inovace výuky prostřednictvím ICT</i>
Číslo projektu	<i>CZ.1.07/1.5.00/34.1009</i>
Číslo DUM	<i>VY_32_INOVACE_267</i>
Autor	<i>Ing. Karel Johanovský</i>
Datum vytvoření	<i>2. dubna 2013</i>
Tematický celek	<i>Programování a vývoj aplikací - řazení</i>
Téma	<i>Select sort</i>
Anotace	<i>Podpora výuky řadících algoritmů</i>
Metodický pokyn	<i>Prezentace s výkladem, časová náročnost 20 minut</i>
Inovace	<i>Podpora vjemu informací u žáka ve fázi expozice . a zejména ve fázi fixace získaných poznatků (dostupný materiál – možnost libovolného počtu opakování)</i>

Obsah

Úvod

Shrnutí minulých přednášek

Select sort

Řešení

Program

Použití

Složitost

Vzorce

Graf



Shrnutí

- V minulých hodinách jsme si představili první řadící algoritmus je tzv. „bublinkové” řazení - bubble sort.
- Princip spočíval v tom, že se postupně procházelo celé pole a porovnávaly se vždy dva sousední prvky.
- Pokud byly v nesprávném pořadí, vyměnily si pozice.
- Jakmile se takto prošlo celé pole, tak nejméně jeden prvek byl na svém místě.
- Poté jsme pole o tento prvek zkrátili a celý postup opakovali znovu.
- Představili jsme si také jeho vylepšené varianty: zarážku, ripple, shake a shuttle sort



Select sort

- To by bylo krátké připomenutí řadícího algoritmus Bubble sort z minulých přednášek.



Select sort

- To by bylo krátké připomenutí řadícího algoritmus Bubble sort z minulých přednášek.
- Dnes je na řadě nový algoritmus nazávaný: řazení výběrem, neboli select sort.



Select sort

- To by bylo krátké připomenutí řadícího algoritmus Bubble sort z minulých přednášek.
- Dnes je na řadě nový algoritmus nazávaný: řazení výběrem, neboli select sort.
- Princip spočívá v tom, že se v poli najde buď maximum, nebo minimum a tento prvek se prohodí s prvkem na posledním, resp. prvním místě.



Select sort

- To by bylo krátké připomenutí řadícího algoritmus Bubble sort z minulých přednášek.
- Dnes je na řadě nový algoritmus nazávaný: řazení výběrem, neboli select sort.
- Princip spočívá v tom, že se v poli najde buď maximum, nebo minimum a tento prvek se prohodí s prvkem na posledním, resp. prvním místě.
- Poté postup opakujeme, s vynecháním tohoto posledního / prvního prvku.



Select sort

- To by bylo krátké připomenutí řadícího algoritmus Bubble sort z minulých přednášek.
- Dnes je na řadě nový algoritmus nazávaný: řazení výběrem, neboli select sort.
- Princip spočívá v tom, že se v poli najde buď maximum, nebo minimum a tento prvek se prohodí s prvkem na posledním, resp. prvním místě.
- Poté postup opakujeme, s vynecháním tohoto posledního / prvního prvku.
- Po proběhnutí každé jedné fáze máme vždy nejméně jeden prvek na správné pozici.

Select sort v JAVĚ

- Naše řešení naprogramujeme jako funkci, která převeze pole a seřadí jej.

```
public static void SelectSort(int[] pole) {  
    int i = 0;  
    int j = 0;  
    for (i = 0; i < (pole.length - 1); i++) {  
        int max_pos = pole.length - 1 - i;  
        for (j = 0; j < pole.length - i; j++) {  
            if (pole[j] > pole[max_pos]) {  
                max_pos = j;  
            }  
        }  
        int tmp = pole[pole.length-1-i];  
        pole[pole.length-1-i] = pole[max_pos];  
        pole[max_pos] = tmp;  
    }  
}
```



Použití

- Poté naši funkci vezmeme a použijeme v metodě main, kterou jsme vytvořili v první přednášce.

```
import java.util.Random;
public class SortingAlg {
    public static void main(String[] args) {
        int velikost = 100;
        Random rd = new Random();
        int pole[] = new int[velikost];
        for (int i = 0; i < pole.length; i++) {
            pole[i] = rd.nextInt(velikost);
            System.out.print(pole[i] + "\t");
        }
        SelectSort(pole);
        for (int i = 0; i < pole.length; i++) {
            System.out.print(pole[i] + "\t");
        }
    }
}
```



Složitost - vzorce

- Nyní se podívejme na složitost algoritmu.

Složitost - vzorce

- Nyní se podívejme na složitost algoritmu.
- Nejprve počet porovnání:
 - $(N-1) + (N-2) + \dots + 2 + 1 = \frac{1}{2} (N^2 - N)$

Složitost - vzorce

- Nyní se podívejme na složitost algoritmu.
- Nejprve počet porovnání:
 - $(N-1) + (N-2) + \dots + 2 + 1 = \frac{1}{2} (N^2 - N)$
- Nyní počet přesunů:
 - $(N-1)$

Složitost - vzorce

- Nyní se podívejme na složitost algoritmu.
- Nejprve počet porovnání:
 - $(N-1) + (N-2) + \dots + 2 + 1 = \frac{1}{2} (N^2 - N)$
- Nyní počet přesunů:
 - $(N-1)$
- Zde už je patrná první velká změna, oproti všem variantám bubble sortu, přesuny prvků jsou pouze ve vnějším for-cyklu a tudíž se provádí pouze N-krát.

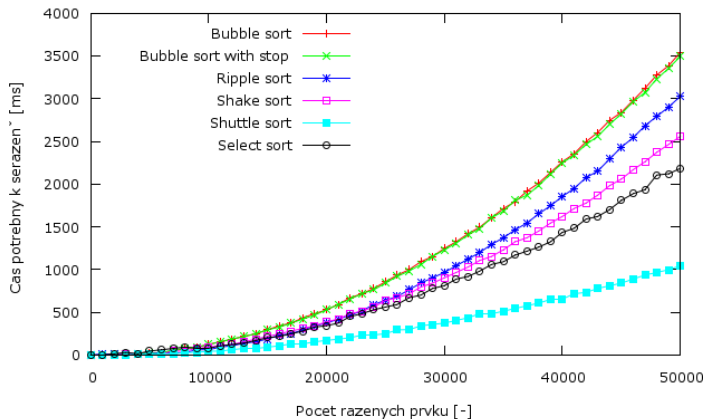


Složitost - vzorce

- Nyní se podívejme na složitost algoritmu.
- Nejprve počet porovnání:
 - $(N-1) + (N-2) + \dots + 2 + 1 = \frac{1}{2} (N^2 - N)$
- Nyní počet přesunů:
 - $(N-1)$
- Zde už je patrná první velká změna, oproti všem variantám bubble sortu, přesuny prvků jsou pouze ve vnějším for-cyklu a tudíž se provádí pouze N-krát.
- Nic méně, z asymptotického hlediska, ale stále platí že nejhorší funkcí je N^2 a musíme tedy konstatovat, že složitost řadícího algoritmu select sort je kvadratická.



Složitost - graf



Obrázek : Složitost select sort



Závěr - co jsme se dozvěděli?

Závěr - co jsme se dozvěděli?

- Zopakovali jsme si princip řadícího algoritmu bubble sort.

Závěr - co jsme se dozvěděli?

- Zopakovali jsme si princip řadícího algoritmu bubble sort.
- Vysvětlili jsme si select sort.

Závěr - co jsme se dozvěděli?

- Zopakovali jsme si princip řadícího algoritmu bubble sort.
- Vysvětlili jsme si select sort.
- Ukázali jsme si jeho realizaci v jazyce Java.

Závěr - co jsme se dozvěděli?

- Zopakovali jsme si princip řadícího algoritmu bubble sort.
- Vysvětlili jsme si select sort.
- Ukázali jsme si jeho realizaci v jazyce Java.
- Spočítali jsme jeho složitost.

Reference



KNUTH, Donald Ervin.

Art of Computer Programming, Volume 3: Sorting and Searching.
Reading, Massachusetts: Addison-Wesley, 1998.
ISBN 0-201-89685-0.



Selection Sort: Sorting Algorithm Animation. [online].

[cit. 2013-03-30]. Dostupné z:

<http://www.sorting-algorithms.com/selection-sort>

- Tento materiál je určen pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.
- Všechna neocitovaná autorská díla jsou dílem autora.
- Všechny neocitované obrázky jsou součástí prostředků použitého výukového software GnuPlot 4.4.0