

Algoritmy a Dátové Štruktúry

Jana Katreniaková

`katreniakova@dcs.fmph.uniba.sk`

Motivácia

- ..
- ..
- ..

Ciele

- Máme neutriedenú postupnosť prvkov v poli, chceme postupnosť utriedenú
- Aké prvky máme? (obmedzené hodnoty, neopakujúce sa prvky, veľa rovnakých)
- Priemerná zložitosť vs. zložitosť v "zlom" prípade (ako často je zle?)

Motivácia

- ..
- ..
- ..

Ciele

- Máme neutriedenú postupnosť prvkov v poli, chceme postupnosť utriedenú
- Aké prvky máme? (obmedzené hodnoty, neopakujúce sa prvky, veľa rovnakých)
- Priemerná zložitosť vs. zložitosť v "zlom" prípade (ako často je zle?)

Triedenia nešpecifikovaných prvkov

Kvadratické triedenia

- Selectionsort
- Insertsort
- Bubblesort

Optimálne triedenia $O(N \cdot \log N)$

- Heapsort
- Mergesort
- Quicksort

Lineárne? triedenia

- Countsort
- Bucketsort

Triedenia nešpecifikovaných prvkov

Kvadratické triedenia

- Selectionsort
- Insertsort
- Bubblesort

Optimálne triedenia $O(N \cdot \log N)$

- Heapsort
- Mergesort
- Quicksort

Lineárne? triedenia

- Countsort
- Bucketsort

Triedenia nešpecifikovaných prvkov

Kvadratické triedenia

- Selectionsort
- Insertsort
- Bubblesort

Optimálne triedenia $O(N \cdot \log N)$

- Heapsort
- Mergesort
- Quicksort

Lineárne? triedenia

- Countsort
- Bucketsort

Stabilita

- Snaha zachovať pôvodné poradie prvkov v prípade ich rovnosti
- ..

Vnútorne vs. vonkajšie

- Vnútorne využívajú iba pole, v ktorom sú zadané
- Výhody a nevýhody

Ďalšie miery zložitosti

- Počet porovnaní
- Počet výmien

Stabilita

- Snaha zachovať pôvodné poradie prvkov v prípade ich rovnosti
- ..

Vnútorne vs. vonkajšie

- Vnútorne využívajú iba pole, v ktorom sú zadané
- Výhody a nevýhody

Ďalšie miery zložitosti

- Počet porovnaní
- Počet výmien

Stabilita

- Snaha zachovať pôvodné poradie prvkov v prípade ich rovnosti
- ..

Vnútorne vs. vonkajšie

- Vnútorne využívajú iba pole, v ktorom sú zadané
- Výhody a nevýhody

Ďalšie miery zložitosti

- Počet porovnaní
- Počet výmien

Myšlienka

- Nájdime najväčší prvok a uložme ho na koniec.
- Potom nájdime najväčší medzi zvyšnými a uložme ho na druhé miesto odzadu atď.

Zložitosť

- Nájdienie najväčšieho prvku z N prvkového poľa $O(N)$
- Presun na posledné miesto $O(1) \Rightarrow$ dostávame menšie pole
- Opakujeme až po pole veľkosti 1
- $O(N) + O(N - 1) + \dots + O(1) = O(N^2)$ VŽDY???

Myšlienka

- Nájdime najväčší prvok a uložme ho na koniec.
- Potom nájdime najväčší medzi zvyšnými a uložme ho na druhé miesto odzadu atď.

Zložitosť

- Nájdienie najväčšieho prvku z N prvkového poľa $O(N)$
- Presun na posledné miesto $O(1) \Rightarrow$ dostávame menšie pole
- Opakujeme až po pole veľkosti 1
- $O(N) + O(N - 1) + \dots + O(1) = O(N^2)$ VŽDY???

MaxSort

MAXINDEX(A,n)

```
1  index ← 0;  
2  for i ← 1 to n - 1  
3  do if ( $A[\textit{index}] < A[\textit{i}]$ )  
4     then index ← i;  
5  return index;
```

MAXSORT(A,n)

```
1  index ← 0;  
2  for kam ← n - 1 to 1  
3  do index ← MAXINDEX(A, kam + 1);  
4     SWAP(index, kam);  
5
```

Myšlienka

- Budeme vymieňať susedné prvky v poli, ak sú v nesprávnom poradí.
- Keď pri prechode poľom nenájdeme žiadne dva prvky v zlom poradí, je pole utriedené.

Zložitosť

- Prechod N prvkového poľa $O(N)$
- Koľkokrát musíme prechod robiť? Kým nebude hotovo...
- $O(N) * O(N - 1) = O(N^2)$ VŽDY???
- Dá sa to nejako zefektívniť?

Myšlienka

- Budeme vymieňať susedné prvky v poli, ak sú v nesprávnom poradí.
- Keď pri prechode poľom nenájdeme žiadne dva prvky v zlom poradí, je pole utriedené.

Zložitosť

- Prechod N prvkového poľa $O(N)$
- Koľkokrát musíme prechod robiť? Kým nebude hotovo...
- $O(N) * O(N - 1) = O(N^2)$ VŽDY???
- Dá sa to nejako zefektívniť?

Bubblesort

```
BUBBLESORT(A,n)
1  hotovo ← false;
2  while (!hotovo)
3  do vymenil ← false;
4     for i ← 1 to n − 1
5     do if  $A[i] < A[i - 1]$ 
6         then SWAP( $A[i], A[i - 1]$ ); vymenil ← true;
7     hotovo ← !vymenil;
```

Myšlienka

- Postupné vkladanie do utriedenej postupnosti.
- Na začiatku máme postupnosť dĺžky 1 (prvok $A[0]$).
Vkladáme $A[i]$ do postupnosti, ktorá je v poli A na miestach $0..i-1$.
- Vkladáme vždy odzadu s využitím poľa na indexoch $0..i$.

Zložitosť

- Vloženie prvku do postupnosti dĺžky i $O(i)$
- Vkladáme postupne prvky $A[1]..A[n-1]$ – prvok $A[i]$ vkladáme do postupnosti dĺžky i
- $O(1) + .. + O(N - 1) = O(N^2)$ VŽDY???

Myšlienka

- Postupné vkladanie do utriedenej postupnosti.
- Na začiatku máme postupnosť dĺžky 1 (prvok $A[0]$).
Vkladáme $A[i]$ do postupnosti, ktorá je v poli A na miestach $0..i-1$.
- Vkladáme vždy odzadu s využitím poľa na indexoch $0..i$.

Zložitosť

- Vloženie prvku do postupnosti dĺžky i $O(i)$
- Vkladáme postupne prvky $A[1]..A[n-1]$ – prvok $A[i]$ vkladáme do postupnosti dĺžky i
- $O(1) + .. + O(N - 1) = O(N^2)$ VŽDY???

Insertsort

```
INSERTSORT(A,n)
1  for  $i \leftarrow 1$  to  $n - 1$ 
2  do
3      $prvok \leftarrow A[i]; kam \leftarrow i;$ 
4     while  $(kam \geq 0) \wedge (A[kam - 1] > prvok)$ 
5     do  $A[kam] \leftarrow A[kam - 1];$ 
6          $kam --;$ 
7      $A[kam] \leftarrow prvok;$ 
```

Myšlienka

- Využijeme haldu – maximum v koreni.
- Najprv haldu musíme z poľa vytvoriť.
- Vždy koreň umiestnime na koniec poľa (vymeníme s posledným) čím haldu o jedna zmenšíme.

Zložitosť

- Vytvorenie haldy $O(N \cdot \log N)$
- Odobratie prvku a upravenie haldy
 $O(1) + O(\log N) = O(\log N)$
- $O(N \cdot \log N) + O(N) * O(\log N) = O(N^2)$ VŽDY???

Myšlienka

- Využijeme haldu – maximum v koreni.
- Najprv haldu musíme z poľa vytvoriť.
- Vždy koreň umiestnime na koniec poľa (vymeníme s posledným) čím haldu o jedna zmenšíme.

Zložitosť

- Vytvorenie haldy $O(N \cdot \log N)$
- Odobratie prvku a upravenie haldy
 $O(1) + O(\log N) = O(\log N)$
- $O(N \cdot \log N) + O(N) * O(\log N) = O(N^2)$ VŽDY???

Heapsort

VYTVORHALDU(A, n)

- 1 **for** $i \leftarrow n/2$ **to** 0
- 2 **do** UPRAVHALDU(A, n, i);

HEAPSORT(A, n)

- 1 VYTVORHALDU(A, n);
- 2 **for** $i \leftarrow n - 1$ **to** 1
- 3 **do** SWAP($A[i], A[1]$);
- 4 $n \leftarrow n - 1$;
- 5 UPRAVHALDU($A, n, 1$);