Algoritmizace: 9. cvičení

Matouš Vrba

15. 11. 2021

Obsah cvičení

- Merge sort
- Heapsort
- Radix sort
- Counting sort

Merge sort (řazení slučováním)

Merge sort (řazení slučováním)

- Rozděl a panuj:
  - Rozdělí množinu na dvě poloviny
  - Seřadí obě poloviny
  - Spojí seřazené podmnožiny

Merge sort (řazení slučováním)

- Rozděl a panuj:
  - Rozdělí množinu na dvě poloviny
  - Seřadí obě poloviny
  - Spojí seřazené podmnožiny
- Asymptotická složitost je $\Theta(n \log(n))$

Příklad 1

Merge sort:

1. lze napsat tak, aby nebyl stabilní
2. je stabilní, protože jeho složitost je $\Theta(n \log(n))$
3. je nestabilní, protože jeho složitost je $\Theta(n \log(n))$
4. jeho složitost je $\Theta(n \log(n))$ právě proto, že je stabilní
5. neplatí ani jedno z předchozích tvrzení

Příklad 1

Merge sort:

✓ lze napsat tak, aby nebyl stabilní
2. je stabilní, protože jeho složitost je $\Theta(n \log(n))$
3. je nestabilní, protože jeho složitost je $\Theta(n \log(n))$
4. jeho složitost je $\Theta(n \log(n))$ právě proto, že je stabilní
5. neplatí ani jedno z předchozích tvrzení

Příklad 2a

V určitém problému je velikost zpracovávaného pole s daty rovna $2N - 2$, kde N charakterizuje velikost problému. Pole se řadí pomocí Merge sortu. Jaká je asymptotická složitost tohoto algoritmu nad uvedeným polem v závislosti na hodnotě N?

Příklad 2a

V určitém problému je velikost zpracovávaného pole s daty rovna $2N - 2$, kde N charakterizuje velikost problému. Pole se řadí pomocí Merge sortu. Jaká je asymptotická složitost tohoto algoritmu nad uvedeným polem v závislosti na hodnotě N?

Asymptotická složitost Merge sortu pro n prvků je $\Theta(n \log(n))$. Dosadíme za $n = 2N - 2$:

$$\begin{aligned}(2N-2)\log(2N-2) &= 2N\log(2N-2) - 2\log(2N-2) \\ &\approx 2N\log(2N) - 2\log(2N) \in \Theta(N\log(N))\end{aligned}$$

Příklad 4a

Merge sort řadí pole šesti čísel 8 1 7 6 4 2. Jaká bude situace v řazeném poli těsně před provedením posledního slévání?

1. 1 2 4 6 7 8
2. 8 7 6 4 2 1
3. 1 7 8 2 4 6
4. 1 2 4 7 8 6
5. 2 4 6 1 7 8

Příklad 4a

Merge sort řadí pole šesti čísel 8 1 7 6 4 2. Jaká bude situace v řazeném poli těsně před provedením posledního slévání?

1. 1 2 4 6 7 8
2. 8 7 6 4 2 1
✓ 1 7 8 2 4 6
4. 1 2 4 7 8 6
5. 2 4 6 1 7 8

Heapsort

- Využívá *haldu*:
  - Binární stromová struktura.
  - Min-halda: Pro každý uzel platí, že má menší klíč než oba jeho potomci.
  - Max-halda: Obdobně...
  - Lze reprezentovat polem – pro uzel na indexu i platí, že jeho levý potomek je na indexu $2i$ a pravý na indexu $2i + 1$ (indexováno od 1).
- Z neseřazeného pole se nejdříve vytvoří halda.
- Iterativně je odstraňován kořen haldy a nahrazován novým pomocí algoritmu opravení haldy.
- Odstraněný kořen je v každém kroku vložen na konec výstupního pole.
- Asymptotická složitost je $O(n \log(n))$.

Příklad 5

Předložíme-li heapsortu vzestupně seřazenou posloupnost délky n, jaká bude složitost celého řazení?

1. $\Theta(n)$, protože heapsort vytvoří haldu v čase $\Theta(n)$
2. $\Theta(n^2)$, protože heapsort vytvoří haldu v čase $\Theta(n^2)$
3. $\Theta(n \log(n))$, protože to je složitost vytvoření haldy
4. $\Theta(n \log(n))$, protože to je složitost zpracování haldy
5. $\Theta(n)$, protože to je složitost vytvoření i zpracování haldy

Příklad 5

Předložíme-li heapsortu vzestupně seřazenou posloupnost délky n, jaká bude složitost celého řazení?

1. $\Theta(n)$, protože heapsort vytvoří haldu v čase $\Theta(n)$
2. $\Theta(n^2)$, protože heapsort vytvoří haldu v čase $\Theta(n^2)$
3. $\Theta(n \log(n))$, protože to je složitost vytvoření haldy

✓ $\Theta(n \log(n))$, protože to je složitost zpracování haldy

5. $\Theta(n)$, protože to je složitost vytvoření i zpracování haldy

Příklad 6

Heapsort:

1. není stabilní, protože halda nemusí být pravidelným stromem
2. není stabilní, protože žádný algoritmus se složitostí $\Theta(n \log(n))$ nemůže být stabilní
3. je stabilní, protože halda je vyvážený strom
4. je stabilní, protože to zaručuje pravidlo haldy
5. neplatí ani jedno z předchozích tvrzení

Příklad 6

Heapsort:

1. není stabilní, protože halda nemusí být pravidelným stromem
2. není stabilní, protože žádný algoritmus se složitostí $\Theta(n \log(n))$ nemůže být stabilní
3. je stabilní, protože halda je vyvážený strom
4. je stabilní, protože to zaručuje pravidlo haldy

✓ neplatí ani jedno z předchozích tvrzení

Příklad 8a

Je dána halda uložená v poli. Proveďte první krok heapsortu (tj. zařazení nejmenšího prvku na jeho definitivní pozici a upravení pole tak, aby opět tvořilo haldu).

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hodnota	1	5	2	17	13	24	9	19	23	22

Příklad 8a

Je dána halda uložená v poli. Proveďte první krok heapsortu (tj. zařazení nejmenšího prvku na jeho definitivní pozici a upravení pole tak, aby opět tvořilo haldu).

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hodnota	1	5	2	17	13	24	9	19	23	22

Např.

index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hodnota	2	5	9	17	13	24	22	19	23	1

Radix sort (přihrádkové řazení)

- Slouží k řazení řetězců totožné délky
- Nejdříve řetězce řadí podle posledního znaku, poté podle předposledního, . . . , až podle prvního
- Asymptotická složitost je $\Theta(k(n + |\Sigma|))$, kde k je délka řetězců, n je počet řetězců a $|\Sigma|$ velikost abecedy

Příklad 2a.2

Následující posloupnost řetězců je nutno seřadit pomocí Radix sortu. Proveďte první průchod daty a napište, jak budou následně data seřazena.

```
IIYI
PIYY
YIII
YPPP
YYYI
PYPP
PIPI
PPYI
```

Příklad 2a.2

Následující posloupnost řetězců je nutno seřadit pomocí Radix sortu. Proveďte první průchod daty a napište, jak budou následně data seřazena.

```
IIYI
PIYY
YIII
YPPP
YYYI
PYPP
PIPI
PPYI
```

```
IIYI
YIII
YYYI
PIPI
PPYI
YPPP
PYPP
PIYY
```

Příklad Radix sort

Seřaďte následující řetězce pomocí Radix sortu.

0	bbc
1	bac
2	bcb
3	aaa
4	aac
5	cbc
6	caa

	z	k
a		
b		
c		

0	
1	
2	
3	
4	
5	
6	

Příklad Radix sort

Obsah polí po prvním průchodu:

0	bbc
1	bac
2	bcb
3	aaa
4	aac
5	cbc
6	caa

	z	k
a	3	6
b	2	2
c	0	5

0	1
1	4
2	-
3	6
4	5
5	-
6	-

Jak budeme pokračovat?

Counting sort

- Vhodný pro řazení velkých polí, které nabývají pouze malého počtu možných hodnot.
- Při průchodu polem se zaznamenávají četnosti jednotlivých prvků do pomocného pole, následně se modifikuje pole četností (na kumulativní součet).
- V dalším kroku se pole prochází zprava doleva, z modifikovaného pole četností lze najít index prvku, tento index se sníží o jedna.
- Asymptotická složitost je $O(n + k)$, kde k je rozsah hodnot.

Příklad 9.2

Uvažujme pole obsahující 1000 navzájem různých čísel v pohyblivé řadové čárce. Counting sort se pro toto pole:

1. hodí, protože toto řazení má lineární složitost.
2. hodí, protože toto řazení má sublineární složitost.
3. hodí, protože čísla lze převést na řetězce.
4. nehodí, protože čísla v poli nemusí být celá.
5. nehodí, protože čísla jsou navzájem různá.

Příklad 9.2

Uvažujme pole obsahující 1000 navzájem různých čísel v pohyblivé řadové čárce. Counting sort se pro toto pole:

1. hodí, protože toto řazení má lineární složitost.
2. hodí, protože toto řazení má sublineární složitost.
3. hodí, protože čísla lze převést na řetězce.
4. nehodí, protože čísla v poli nemusí být celá.

✓ nehodí, protože čísla jsou navzájem různá.

Samostatná práce

- Ve skupinách řešte úlohy 10, 11 a 12 z první prezentace a 13 z druhé prezentace.
- Odpovědi zašlete na e-mail matous.vrba@fel.cvut.cz, s předmětem ALG09.