

# ŘAZENÍ RYCHLEJŠÍ NEŽ KVADRATICKÉ

---

Karel Horák, Petr Ryšavý

20. dubna 2016

Katedra počítačů, FEL, ČVUT

Která z následujících posloupností představuje haldu uloženou v poli?

1. 9 5 4 6 3

2. 5 4 2 3 9

3. 3 8 9 5 6

4. 5 1 8 9 1

5. 1 3 6 5 4

V haldě, jejíž vrchol obsahuje minimální prvek haldy, máme najít prvek s maximálním klíčem. Jaká je asymptotická složitost této akce?

Umíte vyřešit problém lépe (ne nutně asymptoticky) než v  $n$  operacích.

Je dáno  $n$  ( $n \in \mathbb{N}, n \geq 2$ ) navzájem různých celočíselných klíčů a prázdná halda. Všechny klíče vložíme jeden po druhém v náhodném pořadí do dané haldy.

1. Jaká je asymptotická složitost tohoto procesu?
2. Je možné, že pro některé speciální pořadí klíčů bude asymptotická složitost menší nebo větší než v náhodném případě?

Danou haldu s  $n$  prvky máme rozdělit na dvě haldy, tak že každá bude mít  $\frac{n}{2}$ . Předpokládejme, že původní halda je uložena v poli délky  $n$  a nové haldy budou uloženy ve dvou připravených polích délky  $\frac{n}{2}$ .

Navrhněte, jak rozdělit haldu v čase  $\Theta(n)$ .

Pole  $n$  prvků uspořádané v rostoucím pořadí lze považovat za haldu. Z této operace odstraníme standardní operací `deleteTop()` její vrchol. Určete za jakých okolností je možné, aby výsledné pole bylo po uvedené operaci opět celé uspořádané.

Merge sort řadí pole šesti čísel: 8, 1, 7, 6, 4, 2. Jak bude toto pole vypadat před provedením poslední operace merge?

Jaká je asymptotická složitost algoritmu zvaného *4-way merge sort*.  
Algoritmus funguje jako standardní merge sort, pouze pole rozkládá na 4 části místo dvou.



Jaká bude asymptotická složitost řazení, předložíme-li Heap sort-u vzestupně seřazenou posloupnost délky  $n$ .

Je heap sort stabilní řazení? Proč? Pokud není, uměli byste upravit heap sort tak, aby byl stabilní?

Zadanou posloupnost v poli seřadíte ručně pomocí heap sortu. Registrujte stav v poli po každém kroku. Nejprve po každé opravě částečné haldy od prostředku pole směrem k začátku, tj. při první fázi. Potom také po každé opravě haldy a přenesení prvku z vrcholu haldy za konec haldy.

23 29 27 4 28 17 1 24 6 30 19

Vysvětlete, jak je nutno modifikovat Heap sort, aby po jeho skončení pole obsahovalo prvky seřazené vzestupně. Algoritmus musí být stejně rychlý, nejen asymptoticky, a nesmí používat žádné další datové struktury nebo proměnné.

Poskytneme-li Quick Sort-u (Q) a Merge sort-u (M) stejná data k seřazení, platí

1. Q bude vždy asymptoticky rychlejší než M
2. M bude vždy asymptoticky rychlejší než Q
3. někdy může být Q asymptoticky rychlejší než M
4. někdy může být M asymptoticky rychlejší než Q
5. Q i M budou vždy asymptoticky stejně rychlé

Pomocí Radix sort-u řadíme pole  $n$  řetězců, každý řetězec má kladnou délku  $k$ . Jaká je asymptotická složitost tohoto řazení?

Následující posloupnost řetězců je nutno seřadit pomocí Radix Sortu (přihrádkového řazení). Proveďte první průchod algoritmu danými daty a napište, jak budou po tomto prvním průchodu seřazena.

AACA BACC CAAA CBBB CCCA BCBB BABA BBKA

Pole  $A$  obsahuje téměř seřazené řetězce, pole  $B$  obsahuje řetězce stejné délky, ale zcela neseřazené. Radix sort seřadí asymptoticky

1.  $A$  rychleji než  $B$
2.  $B$  rychleji než  $A$
3.  $A$  stejně rychle jako  $B$ , použije více paměti pro řazení  $A$
4.  $A$  stejně rychle jako  $B$ , použije více paměti pro řazení  $B$
5.  $A$  stejně rychle jako  $B$  a použití paměti bude stejné



Radix sort řadí dané pole řetězců. Napište, budou řetězce rozděleny do přihrádek po prvním průchodu algoritmu tj, po seřazení podle posledního znaku.

dda, bab, ddc, aaa, bcd, dbc, bbb, add, ccd, dab, bbc

Každé kladné číslo typu `int` lze interpretovat jako posloupnost čtyř znaků reprezentujících jeho zápis v soustavě o základu 256.

Jak je nutno modifikovat Radix sort, aby mohl řadit pole takto interpretovaných celých čísel?

Bude to časově/paměťově výhodné? Pro jak velká pole?

Uvažujme pole obsahující 1000 navzájem různých čísel v pohyblivé řádové čárce. Hodí se counting sort pro seřazení tohoto pole?