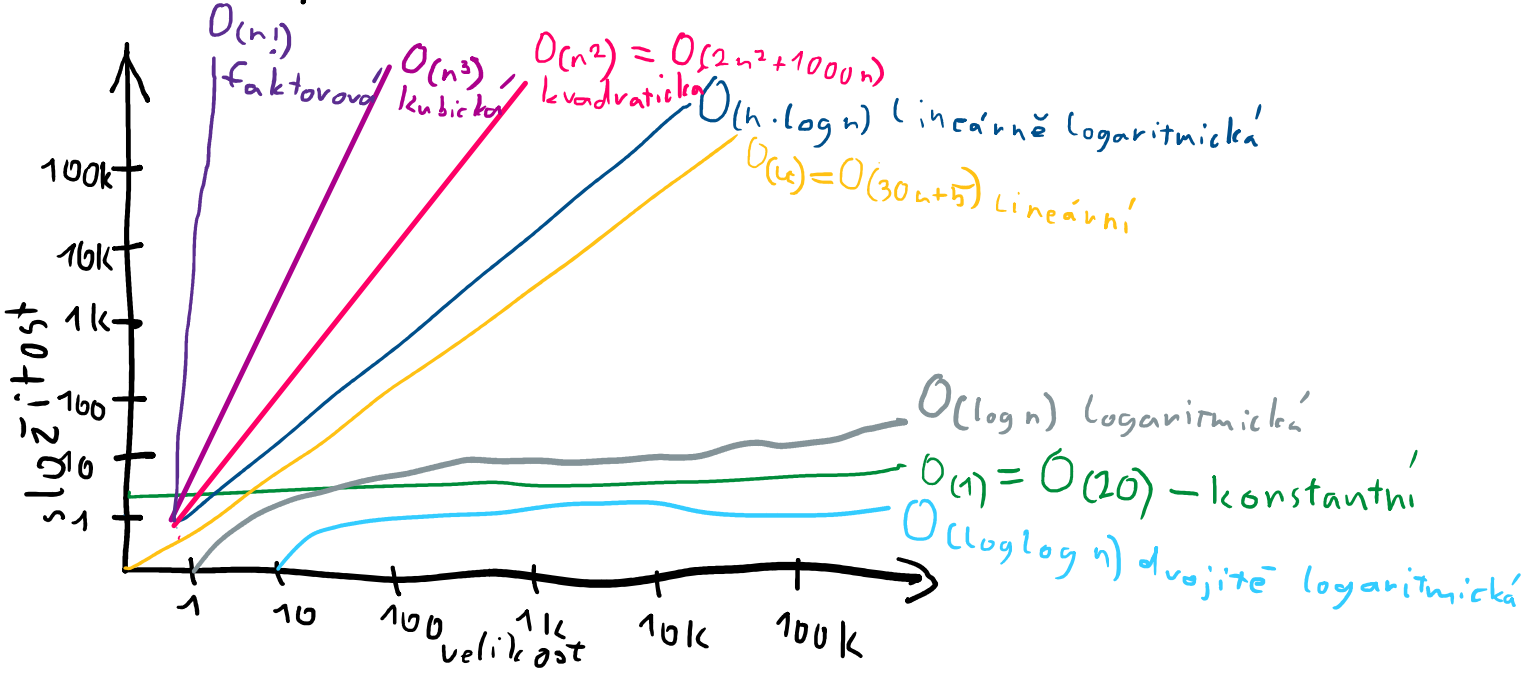
Informatika 2. ročník

- porovnání asymptomických složitostí – třídy
- rychlost růstu časové nebo paměťové složitosti v závislosti na velikosti vstupu – počet položek



- nezáleží na absolutní hodnotě složitosti, ale na rychlosti růstu

v bodě → směrnice tečny

2.hodina 18.9 2023

pondělí 11. září 2023 14:02

Řadící algoritmy

- motivace - v seřazené posloupnosti může člověk nebo stroj rychleji vyhledávat (nižší třída složitosti)
  - pro rychlé vyhledávání je nutné nejdříve hodnoty efektivně seřadit (s nejnižší složitostí)
- cíl - pořadí podle zadané podmínky
  - (např. podle jména, věku, ...) => podle atributu (vlastností)
  - číselně nebo lexiko-graficky (abecedně), vzestupně nebo sestupně
- vstup - datová struktura obsahující položky
  - lineární = pole (array), seznam (list)
  - stromový = strom (tree), seřazený strom -> halda (heap)
- řazení - každé položce, kterou řadíme je přiřazena klíčem (key) hodnota, kterou použijeme pro řazení
  - let osoba = { jmeno: "Jan", prijmeni: "Novak", vek: "18"}

    osoba[prijmeni] = "Novak"

  => určí hodnotu položky ve výstupu
- výstup - přeuspořádaný vstup - permutace
  - počet permutací je n!, n je počet položek
- třídící algoritmus - zařazení položky podle hodnoty klíče do určité skupiny
  - např. věk ≤ 18 ; věk > 18
  - součást rychlých řadících algoritmů
- druhy řazení - číselné (numerické), abecední (lexikografické)
- typy řazení - náhodné (random)

- druhy řazení - číselné (numerické), abecední (lexikografické)

- typy řazení - náhodné (random)

  - vzestupně (ascending)

  - sestupně (descending)

3.hodina 25.9. 2023

pondělí 25. září 2023 10:53

Faktory ovlivňující rychlost řazení

- zvolení algoritmu - třída výpočetní složitosti $O(n!)$, $O(n^2)$, $O(n \log n)$
  - $O(n!)$ náhodné
  - $O(n^2)$ pomalé
  - $O(n \log n)$ rychlé
- řazené hodnoty - počet a charakteristika
  - počet od 1 - miliard, n
  - charakteristika - několik unikátních (- např. známky 1-5)
    - typ hodnot: - mnoho unikátních (- např. ceny zboží)
      - malé rozpětí (- např. odchylky 0mm-20mm)
      - velké rozpětí (- např. vzdálenosti)
    - seřazenost - náhodná, částečná, téměř úplná, opačná
- datová struktura - průměrná a maximální složitost operací
  - pole (array) - přístup $\Theta(1)$ a $O(1)$, vložení a odstranění $\Theta(n)$ a $O(n)$
  - seznam (list) - vkládání a odstraňování $\Theta(1)$ a $O(1)$, přístup $\Theta(n)$ a $O(n)$
  - binární strom (tree) - přístup, odstranění, vložení $\Theta(\log n)$ a $O(\log n)$
  - halda (heap) - vyhledání minima (maxima) $\Theta(1)$ a $O(1)$, správa haldy $O(\log n)$

- hardware - fyzické komponenty PC
  - paměť - přístup: - náhodný = čas není ovlivněn pozicí dat v paměti (čtení, zápis)
    - např. = registry, RAM, SSD
    - sekvenční = čas výrazně ovlivňuje umístění dat
    - např = HDD, DVD, CD, mag. páska
  - rychlost: - čas čtení (zápisu) v závislosti na velikosti dat (MB/s, GB/s)
  - přístupová doba: - čas zpřístupnění umístění dat
  - výpočetní jednotky - porovnání (počítání) hodnot, operace s dat. strukturou
    - paralelní (souběžné) zpracování, nebo sériové (sekvenční)

- fyzický procesor = obsahuje log. procesory a cache (L2, L3)
- logický procesor = obsahuje jádra a cache (L1)
- jádro procesoru = registry a ALU
- specializovaný HW = ASIC (Application Specific Intergrated Circuit)

4.hodina 2.10. 2023

pondělí 2. října 2023 11:51

Typy řadících algoritmů

- pracovní paměť: - různé typy algoritmů mohou využívat různou velikost paměti
  - velikost paměti nutná pro vlastní uložení položek se nevažuje
  - „in-place": - potřebují pro svoji činnost O(1) pomocné paměti (pomocné proměnné)
    - přímo transformují zadaný vstup i s minimem další potřební paměti
    - v opačném případě je potřeba více volné paměti pro funkci algoritmu
  - rekurzivní: - z pravidla potřebují O(log n) paměti a lze je považovat za „in-place"
    - paměť je použita pro zásobník volání (call stack)
  - ostatní: - potřebné množství paměti se liší dle charakteristik řazených položek
- implementace = realizace algoritmu
  - iterativní: - použití cyklů (for, while, ...)
    - nepoužívá rekurzi
  - rekurzivní: - algoritmus používá sám svoji definici (funkci) se zkracujícím se vstupem
    - zásobník volání O(log n) paměti
    - rychlejší než in-place (iterativní)
    - např. quick sort
  - kombinované: - iterace, rekurze, např. merge sort
- porovnávání
  - přímé: - vždy dvě vybrané řazené hodnoty
    - relační operátory <, >, <=, =>
    - výsledek — true (správné pořadí porovnávaných položek)
      false (nesprávné/opačné pořadí)
      => položky prohodíme (swap) do správného pořadí
    - např. u buble, selection, insertion => počet porovnání O(n²)

porovnáváme [illegible] do správného pořadí

- např. u bubble, selection, insertion => počet porovnání $O(n^2)$

- nepřímé: - např. třídění položek do kategorií

- lepší výsledek než $O(n \log n)$ - Quick, merge

- např. (counting sort $\sim O(n)$

- řazené hodnoty: - čísla jako řetězce 100 = "100"

- pokud je algoritmus <u>stabilní</u> lze řadit opakovaně

podle => jednotek, desítek, stovek, ...