

Složitosti základních operací B^+ stromu

Radim Bača

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky



Složitosti základních operací B^+ stromu

Radim Bača

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky



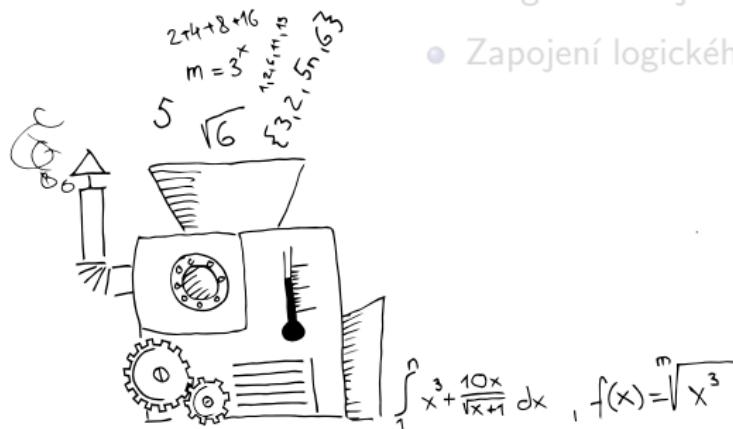
Obsah

- Algoritmus
- Složitost
- Vyhledávání
- B^+ strom

Algoritmus

Zadání algoritmu:

- Slovní popis
- Program v nějakém programovacím jazyce
- Zapojení logického obvodu



$$\int_1^3 x^3 + \frac{10x}{\sqrt{x+1}} dx, \quad f(x) = \sqrt[3]{x^3}$$

Algoritmus

Zadání algoritmu:

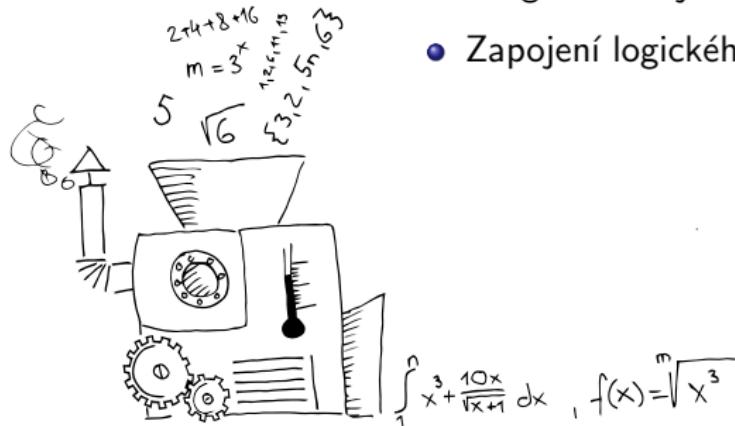
- Slovní popis
- Program v nějakém programovacím jazyce
- Zapojení logického obvodu



Algoritmus

Zadání algoritmu:

- Slovní popis
- Program v nějakém programovacím jazyce
- Zapojení logického obvodu



Problém

Zadání problému:

- Množina vstupů
- Množina výstupů
- Vzájemný vztah mezi vstupy a výstupy

Problém

Zadání problému:

- Množina vstupů
- Množina výstupů
- Vzájemný vztah mezi vstupy a výstupy

Problém

Zadání problému:

- Množina vstupů
- Množina výstupů
- Vzájemný vztah mezi vstupy a výstupy

Algoritmus

Otázky, jež nás trápí:

- Jak popsat kvalitu algoritmu?
- Jak rozhodnout, který z algoritmů je lepší?

Algoritmus

Otázky, jež nás trápí:

- Jak popsat kvalitu algoritmu?
- Jak rozhodnout, který z algoritmů je lepší?

Složitost

- Složitost algoritmu popisujeme funkcí $f(n)$, kde n je velikost vstupu
- Na složitost můžeme nahlížet z mnoha pohledů:
 - Časová složitost
 - Prostorová složitost
 - Finanční složitost

Složitost

- Složitost algoritmu popisujeme funkcí $f(n)$, kde n je velikost vstupu
- Na složitost můžeme nahlížet z mnoha pohledů:
 - Časová složitost
 - Prostorová složitost
 - Finanční složitost

Složitost

- Složitost algoritmu popisujeme funkcí $f(n)$, kde n je velikost vstupu
- Na složitost můžeme nahlížet z mnoha pohledů:
 - Časová složitost
 - Prostorová složitost
 - Finanční složitost

Složitost

- Složitost algoritmu popisujeme funkcí $f(n)$, kde n je velikost vstupu
- Na složitost můžeme nahlížet z mnoha pohledů:
 - Časová složitost
 - Prostorová složitost
 - Finanční složitost

Příklady složitosti

- Mějme algoritmus zpracovávající vstup o velikosti n
- Algoritmus provede $f(n)$ operací, přičemž jedna operace trvá $1 \mu\text{s}$

	10	20	40	100	200	500	1000
n	$10 \mu\text{s}$	$20 \mu\text{s}$	$40 \mu\text{s}$	$0,1 \text{ ms}$	$0,2 \text{ ms}$	$0,5 \text{ ms}$	1 ms
$n \log(n)$	$10 \mu\text{s}$	$26 \mu\text{s}$	$64 \mu\text{s}$	$0,2 \text{ ms}$	$0,46 \text{ ms}$	$1,35 \text{ ms}$	3 ms
n^2	$0,1 \text{ ms}$	$0,4 \text{ ms}$	$1,6 \text{ ms}$	10 ms	40 ms	250 ms	1 s
n^3	1 ms	8 ms	64 ms	1 s	8 s	2 min	16.7 min
2^n	1 ms	1 s	$12,75 \text{ dní}$	$40 \cdot 10^{15} \text{ let}$	-	-	-
$n!$	$3,6 \text{ s}$	$1,8 \cdot 10^6 \text{ let}$	$6,2 \cdot 10^{35} \text{ let}$	-	-	-	-

TABLE: Příklady složitosti

Příklady složitosti

- Mějme algoritmus zpracovávající vstup o velikosti n
- Algoritmus provede $f(n)$ operací, přičemž jedna operace trvá $1 \mu\text{s}$

	10	20	40	100	200	500	1000
n	$10 \mu\text{s}$	$20 \mu\text{s}$	$40 \mu\text{s}$	$0,1 \text{ ms}$	$0,2 \text{ ms}$	$0,5 \text{ ms}$	1 ms
$n \log(n)$	$10 \mu\text{s}$	$26 \mu\text{s}$	$64 \mu\text{s}$	$0,2 \text{ ms}$	$0,46 \text{ ms}$	$1,35 \text{ ms}$	3 ms
n^2	$0,1 \text{ ms}$	$0,4 \text{ ms}$	$1,6 \text{ ms}$	10 ms	40 ms	250 ms	1 s
n^3	1 ms	8 ms	64 ms	1 s	8 s	2 min	16.7 min
2^n	1 ms	1 s	$12,75 \text{ dní}$	$40 \cdot 10^{15} \text{ let}$	-	-	-
$n!$	$3,6 \text{ s}$	$1,8 \cdot 10^6 \text{ let}$	$6,2 \cdot 10^{35} \text{ let}$	-	-	-	-

TABLE: Příklady složitosti

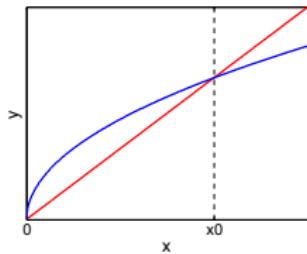
Příklady složitosti

- Mějme algoritmus zpracovávající vstup o velikosti n
- Algoritmus provede $f(n)$ operací, přičemž jedna operace trvá $1 \mu\text{s}$

	10	20	40	100	200	500	1000
n	$10 \mu\text{s}$	$20 \mu\text{s}$	$40 \mu\text{s}$	$0,1 \text{ ms}$	$0,2 \text{ ms}$	$0,5 \text{ ms}$	1 ms
$n \log(n)$	$10 \mu\text{s}$	$26 \mu\text{s}$	$64 \mu\text{s}$	$0,2 \text{ ms}$	$0,46 \text{ ms}$	$1,35 \text{ ms}$	3 ms
n^2	$0,1 \text{ ms}$	$0,4 \text{ ms}$	$1,6 \text{ ms}$	10 ms	40 ms	250 ms	1 s
n^3	1 ms	8 ms	64 ms	1 s	8 s	2 min	16.7 min
2^n	1 ms	1 s	$12,75 \text{ dní}$	$40 \cdot 10^{15} \text{ let}$	-	-	-
$n!$	$3,6 \text{ s}$	$1,8 \cdot 10^6 \text{ let}$	$6,2 \cdot 10^{35} \text{ let}$	-	-	-	-

TABLE: Příklady složitosti

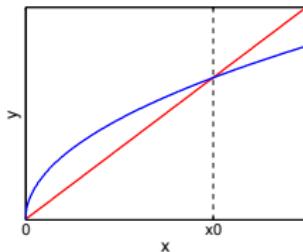
Asymptotická složitost



- Snažíme se popsat obecné vlastnosti funkce složitosti (především odhad rychlosti růstu)
- Zanedbáváme konstanty a méně významné členy
- $f(n) \in O(g(n))$ právě tehdy, když

$$(\exists C > 0) (\exists x_0) (\forall x > x_0) : |f(x)| \leq C |g(x)|$$

Asymptotická složitost



- Snažíme se popsat obecné vlastnosti funkce složitosti (především odhad rychlosti růstu)
- Zanedbáváme konstanty a méně významné členy
- $f(n) \in O(g(n))$ právě tehdy, když

$$(\exists C > 0) (\exists x_0) (\forall x > x_0) : |f(x)| \leq C |g(x)|$$

Definice problému

- V indexu máme uloženy dvojice (klíč, hodnota)
- Základní operace (problémy):
 - Vyhledávání - vyhledání hodnoty se zadaným klíčem
 - Vkládání - vložení nové dvojice

Definice problému

- V indexu máme uloženy dvojice (klíč, hodnota)
- Základní operace (problémy):
 - Vyhledávání - vyhledání hodnoty se zadaným klíčem
 - Vkládání - vložení nové dvojice

Definice problému

- V indexu máme uloženy dvojice (klíč, hodnota)
- Základní operace (problémy):
 - Vyhledávání - vyhledání hodnoty se zadaným klíčem
 - Vkládání - vložení nové dvojice

Definice problému

- V indexu máme uloženy dvojice (klíč, hodnota)
- Základní operace (problémy):
 - Vyhledávání - vyhledání hodnoty se zadaným klíčem
 - Vkládání - vložení nové dvojice

Realita počítače

- Dvojice jsou uloženy na disku
- Data jsou z disku čtena po blocích (i když čteme pouze jednu dvojici)
 - Kapacita disku je řádově větší
 - Data musí být dostupná i po restartu počítače
- Složitost určujeme s ohledem na počet přístupů k jednotlivým blokům - I/O

Realita počítače

- Dvojice jsou uloženy na disku
- Data jsou z disku čtena po blocích (i když čteme pouze jednu dvojici)
 - Kapacita disku je řádově větší
 - Data musí být dostupná i po restartu počítače
- Složitost určujeme s ohledem na počet přístupů k jednotlivým blokům - I/O

Realita počítače

- Dvojice jsou uloženy na disku
- Data jsou z disku čtena po blocích (i když čteme pouze jednu dvojici)
 - Kapacita disku je řádově větší
 - Data musí být dostupná i po restartu počítače
- Složitost určujeme s ohledem na počet přístupů k jednotlivým blokům - I/O

Realita počítače

- Dvojice jsou uloženy na disku
- Data jsou z disku čtena po blocích (i když čteme pouze jednu dvojici)
 - Kapacita disku je řádově větší
 - Data musí být dostupná i po restartu počítače
- Složitost určujeme s ohledem na počet přístupů k jednotlivým blokům - I/O

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

Jenoduchá řešení

C - kapacita bloku

$N = n/C$ - počet bloků

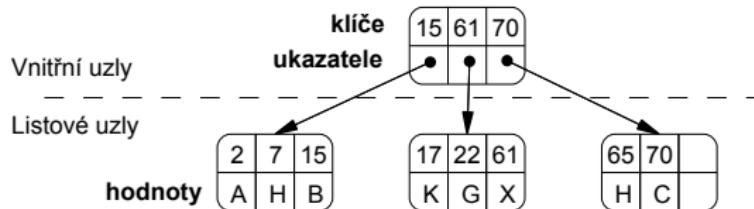
Pole

- Vyhledávání - $O(N)$
- Vkládání - $O(1)$

Setříděné pole

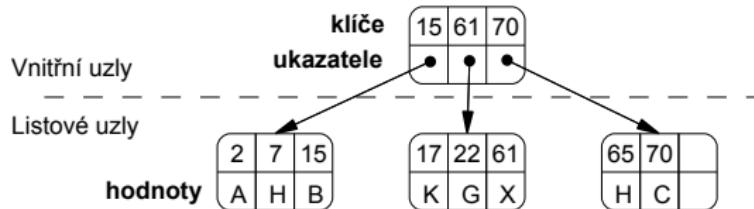
- Vyhledávání - $O(\log_2 N)$
- Vkládání - $O(N)$

B⁺strom



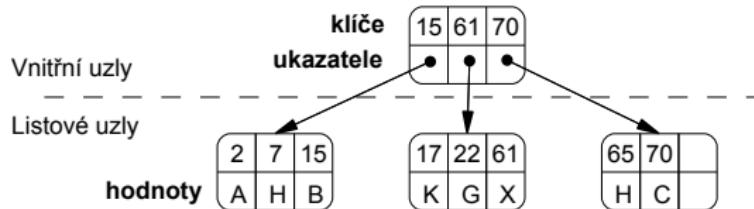
- Hierarchická datová struktura, vyvážený strom
- Uzly vnitřní úrovně se odkazují na uzly nižší úrovně
- Uzly na nejnižší úrovni obsahují hodnoty
- Každý uzel obsahuje setřízené klíče
- Počet klíčů je v rozsahu od $\lceil C/2 \rceil$ až do C

Vyhledávání



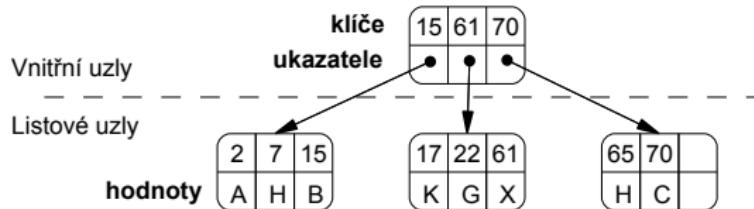
- Postupujeme od kořene k listům
- V každém uzlu nalezneme podle klíče další uzel
- I/O složitost je dána výškou stromu - $\log_C(n/\eta)$
- η - zaplnění stromu

Vyhledávání



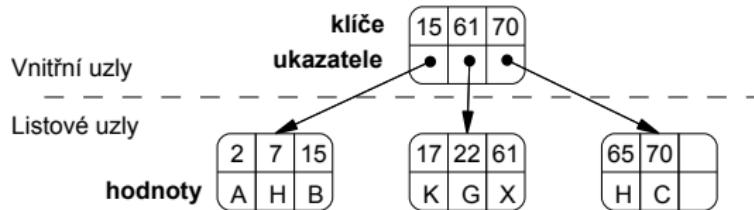
- Postupujeme od kořene k listům
- V každém uzlu nalezneme podle klíče další uzel
- I/O složitost je dána výškou stromu - $\log_C(n/\eta)$
- η - zaplnění stromu

Vyhledávání



- Postupujeme od kořene k listům
- V každém uzlu nalezneme podle klíče další uzel
- I/O složitost je dána výškou stromu - $\log_C(n/\eta)$
- η - zaplnění stromu

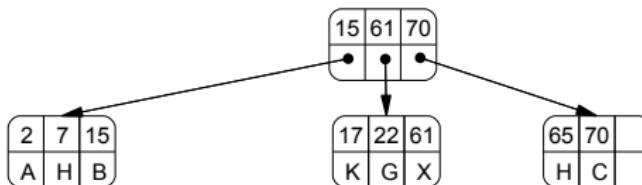
Vyhledávání



- Postupujeme od kořene k listům
- V každém uzlu nalezneme podle klíče další uzel
- I/O složitost je dána výškou stromu - $\log_C(n/\eta)$
- η - zaplnění stromu

Vkládání

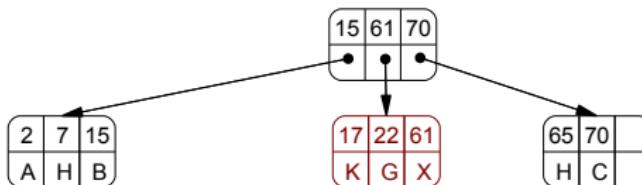
(24, J)



- Nalezneme listový uzel, do kterého budeme vkládat

Vkládání

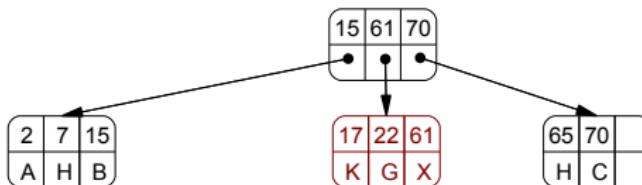
(24, J)



- Nalezneme listový uzel, do kterého budeme vkládat

Vkládání

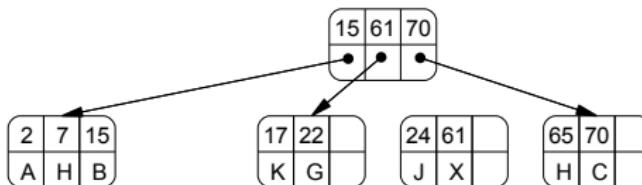
(24, J)



- Nalezneme listový uzel, do kterého budeme vkládat
- 1. Máme dosatek místa, vložíme - $\log_C(n/\eta)$

Vkládání

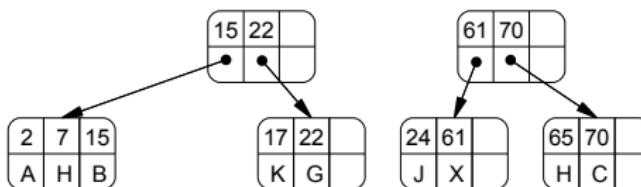
(24, J)



- Nalezneme listový uzel, do kterého budeme vkládat
- 1. Máme dosatek místa, vložíme - $\log_C(n/\eta)$
- 2. Můsíme štěpit

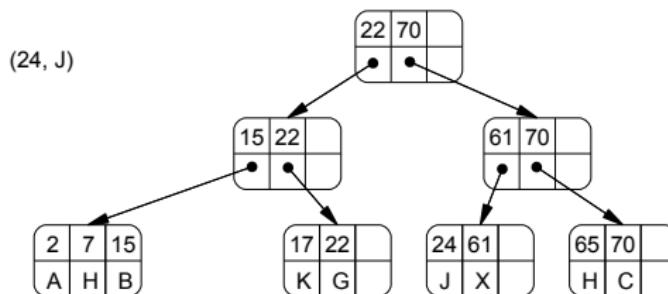
Vkládání

(24, J)



- Nalezneme listový uzel, do kterého budeme vkládat
- 1. Máme dosatek místa, vložíme - $\log_C(n/\eta)$
- 2. Můsíme štěpit
- 2. Rekurzivně postupujeme nahoru

Vkládání



- Nalezneme listový uzel, do kterého budeme vkládat
- 1. Máme dosatek místa, vložíme - $\log_C(n/\eta)$
- 2. Můsíme štěpit
- 2. Rekurzivně postupujeme nahoru
- 2. V případě nutnosti zvedáme výšku stromu - $2 * \log_C(n/\eta) + 1$

η - zaplnění

- Důležitý parametr indexu
- Rozsahové dotazy: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?

η - zaplnění

- Důležitý parametr indexu
- *Rozsahové dotazy*: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?

η - zaplnění

- Důležitý parametr indexu
- *Rozsahové dotazy*: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?

η - zaplnění

- Důležitý parametr indexu
- Rozsahové dotazy: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?

η - zaplnění

- Důležitý parametr indexu
- Rozsahové dotazy: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?
- Vyhledávání volného místa u sousedů (ukazatele na sousedy)
- Nejčastěji tvoříme tři uzly ze dvou

η - zaplnění

- Důležitý parametr indexu
- Rozsahové dotazy: hledáme hodnoty všech dvojic s klíčem spadajícím do intervalu $\langle k_1, k_2 \rangle$
- Závisí na několika parametrech:
 - Frekvence vkládání, mazání a úprav klíčů
 - Pořadí vkládaných klíčů
- Dalo by se zaplnění ovlivnit algoritmem?
- Vyhledávání volného místa u sousedů (ukazatele na sousedy)
- Nejčastěji tvoříme tři uzly ze dvou

Díky za pozornost!

