

Záměna paměťové a časové složitosti

doc. Mgr. Jiří Dvorský, Ph.D.

Stav prezentace ke dni 28. dubna 2024

Katedra informatiky

Fakulta elektrotechniky a informatiky

VŠB – TU Ostrava



Záměna paměťové a časové složitosti

B-stromy

Vyhledávání klíče v B-stromu

Vkládání klíče do B-stromu

Smazání klíče z B-stromu

Záměna paměťové a časové složitosti

B-stromy

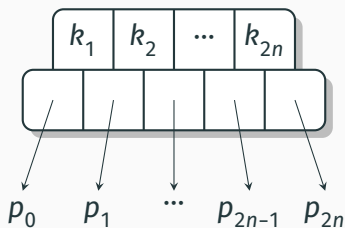
- Zpracování velkého množství strukturovaných záznamů (lze je identifikovat jednoznačným klíčem), které přesahuje dostupnou operační paměť.
- Data musí být uložena ve vnější paměti, tzv. „na disku“.
- Disk nabízí jen sekvenční soubor.
- Hledáme datovou strukturu, která umožní v takovém souboru efektivně vyhledávat, vkládat a mazat záznamy.
- Odpovědí je záměna paměťové složitosti za časovou, jinak řečeno zvýšíme paměťovou složitost (obětujeme extra paměť navíc), abychom snížili časovou složitost operací.

B-strom řádu n je $(2n + 1)$ -ární strom, který splňuje následující kritéria:

1. Každá stránka obsahuje nejvýše $2n$ klíčů.
2. Každá stránka, s výjimkou kořenové obsahuje alespoň n klíčů.
3. Každá stránka je buď listovou stránkou, tj. nemá žádné potomky, nebo má $m + 1$ potomků, kde m je aktuální počet klíčů ve stránce.
4. Všechny listové stránky jsou na stejné úrovni. Jinak řečeno strom je dokonale vyvážený.

Publikováno Rudolfem Bayerem v roce 1972 [6].

B-stromy – schéma stránky

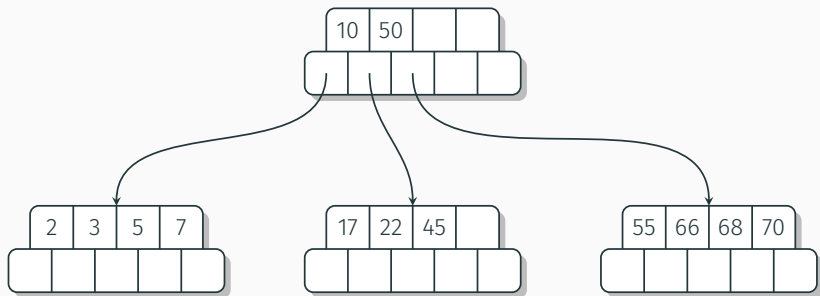


- Uzly v B-stromu se tradičně nazývají **stránky** (pages).
- Počet klíčů ve stránce kolísá od n do $2n$, výjimku tvoří kořen stromu.
- Klíče ve stránce jsou seříděné, tedy
$$k_1 \leq k_2 \leq \dots \leq k_{2n}.$$
- Pro klíče v podstromech na které odkazují ukazatelé p_0, \dots, p_{2n} platí

$$K_{p_0} \leq k_1 \leq K_{p_1} \leq k_1 \leq \dots \leq K_{p_{2n-1}} \leq k_{2n} \leq K_{p_{2n}},$$

kde K_{p_i} je množina všech klíčů v podstromu s kořenem ve stránce na kterou ukazuje p_i .

B-stromy – ukázka



- Z definice je zřejmé, že B-strom není vždy zcela zaplněn. Faktor zaplnění kolísá od 50 % do 100 %.
- Volné místo ve stromu umožňuje snadné vkládání dalších klíčů.
- Díky stromové struktuře lze operace vyhledávání, vkládání a mazání klíčů v B-stromu provádět s logaritmickou časovou složitostí.
- Algoritmy pro B-stromy jsou zobecněním algoritmů pro binární vyhledávací stromy.

B-stromy – alternativní definice, varianty

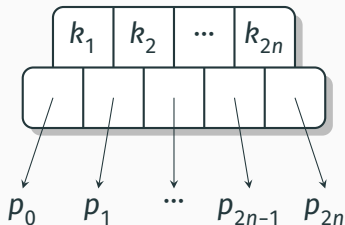
- Výše uvedená definice připouští jen B-stromy s max. kapacitou $2n$ klíčů, tj. sudým číslem.
- Max. kapacita může být ale libovolná, i lichá.
- Některé definice označují pomocí čísla n max. kapacitu.
- Počet klíčů ve stránce tak kolísá od $\left\lceil \frac{n}{2} \right\rceil$ do n .
- Cílem naší definice je snadná pochopitelnost principů fungování B-stromu a jednoduchý zápis.
- Někdy se lze setkat i s definicí, kde číslo n označuje max. počet potomků nikoliv počet klíčů ve stránce.

- B⁺-strom**
 - všechny klíče jsou uloženy pouze v listech
 - listy jsou navzájem propojeny odkazy – rychlejší práce se souvislými úseky klíčů, „najdi všechny klíče mezi 100 a 200“
 - v knize od Levitina [1] je jako B-strom popsána právě tato varianta.
- B^{*}-strom**
 - stránka musí být zaplněna minimálně ze dvou třetin,
 - při vkládání klíče do zaplněné stránky jsou klíče nejprve přesouvány mezi sourozenci,
 - dochází k menšímu počtu štěpení stránek.

B-stromy – vyhledávání klíče x

1. Na počátku algoritmu označíme kořenovou stránku za aktuální stránku P .
2. Pokud stránka P neexistuje, vyhledávání končí neúspěchem.
3. Jinak předpokládejme, že ve stránce P je m klíčů k_1, \dots, k_m a odpovídající ukazatelé na potomky p_0, \dots, p_m . Potom:
 - 3.1 Pokud $x = k_i$, pro některé $1 \leq i \leq m$, pak vyhledání končí úspěchem.
 - 3.2 Jestliže $x < k_1$, potom $P = p_0$ a zpět k bodu 2.
 - 3.3 Jestliže $x > k_m$, potom $P = p_m$ a zpět k bodu 2.
 - 3.4 Jinak nalezneme takové i , $1 \leq i \leq m$, pro které platí, že $k_i < x < k_{i+1}$. Potom $P = p_i$ a a zpět k bodu 2.

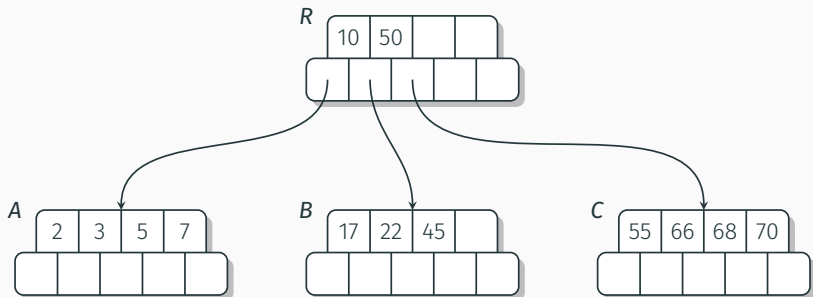
Příklady operací s B-stromem



V ukázkách budeme používat B-strom pro $n = 2$, tzn. každá stránka obsahuje minimálně 2 a maximálně 4 klíče.

A dále každá stránka odkazuje na minimálně 3 a maximálně 5 potomků. Výjimku tvoří kořen stromu.

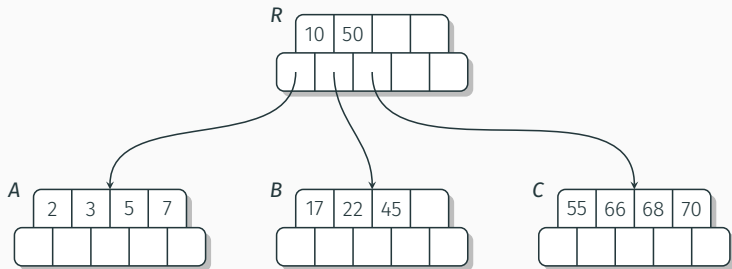
Příklad vyhledávání v B-stromu – vyhledání klíče 50



Postup

1. Vyhledávání zahájíme v kořeni R , tedy $P = R$.
2. Stránka P existuje, pokračujeme dalším bodem.
3. Protože $x = k_2$ hledání končí úspěchem.

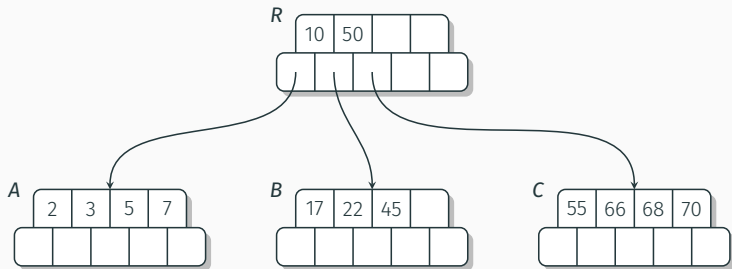
Příklad vyhledávání v B-stromu – vyhledání 3



Postup

1. Vyhledávání zahájíme v kořeni R , tedy $P = R$.
2. Stránka P existuje, pokračujeme dalším bodem.
3. Protože $x < k_1$, pak $P = p_0 = A$.
4. Stránka P existuje, pokračujeme dalším bodem.
5. Protože $x = k_2$ hledání končí úspěchem.

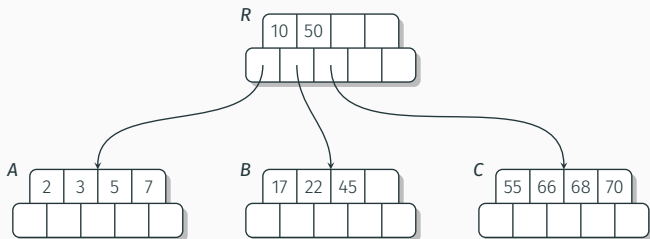
Příklad vyhledávání v B-stromu – vyhledání 45



Postup

1. Vyhledávání zahájíme v kořeni R , tedy $P = R$.
2. Stránka P existuje, pokračujeme dalším bodem.
3. Protože $k_1 < x < k_2$, pak $P = p_1 = B$.
4. Stránka P existuje, pokračujeme dalším bodem.
5. Protože $x = k_3$ hledání končí úspěchem.

Příklad vyhledávání v B-stromu – vyhledání 57



Postup

1. Vyhledávání zahájíme v kořeni R , tedy $P = R$.
2. Stránka P existuje, pokračujeme dalším bodem.
3. Protože $x > k_2$, pak $P = p_2 = C$.
4. Stránka P existuje, pokračujeme dalším bodem.
5. Protože $k_1 < x < k_2$, pak $P = p_1 = \text{null}$.
6. Protože P neexistuje, hledání končí **neúspěchem**.

B-stromy – vložení klíče x

1. Nejprve je nutné určit, pomocí algoritmu vyhledávání, listovou stránku L kam bude klíč x vložen.
2. Mohou nastat dva případy:
 - Stránka L není zcela zaplněna – klíč x je vložen do stránky tak, aby zůstalo zachováno uspořádání klíčů.
 - Stránka L je zcela zaplněna, potom
 - 2.1 klíč x zatřídíme (např. v pomocném poli) mezi klíče ze stránky L tak, aby zůstalo zachováno uspořádání klíčů. Dostaneme posloupnost $2n + 1$ klíčů $k'_1 < k'_2 < \dots < k'_{2n+1}$
 - 2.2 vytvoříme novou stránku P , se stejným rodičem R jako L
 - 2.3 distribuce klíčů do stránek

Klíče	Akce
k'_1, \dots, k'_n	ponechat ve stránce L
k'_n	vložit do rodičovské stránky R
$k'_{n+2}, \dots, k'_{2n+1}$	vložit do nové stránky P

B-stromy – algoritmus vkládání, poznámky

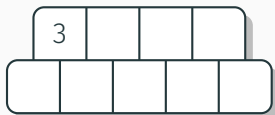
- Proces vytvoření nové stránky a přerozdělení klíčů nazýváme **štěpení stránky**.
- Vložením klíče k'_n do rodičovské stránky R dojde ke zvýšení počtu klíčů v této stránce a tím i k nárůstu počtu odkazů na potomky této stránky. Bez přesunu k'_n by ve stránce R chyběl volný odkaz pro připojení stránky P .
- Vložení klíče k'_n do stránky R se realizuje stejným algoritmem jako vložení klíče x do L . Vložení k'_n může způsobit rozštěpení stránky R .
- Štěpení stránek může vést až k vytvoření nového kořene celého stromu, což je jediný způsob jak B-strom může zvýšit svoji výšku.

Příklad vkládání do B-stromu

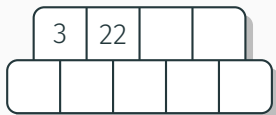
- V tomto rozsáhlejším ukázkovém příkladu budeme postupně budovat B-strom se stejnými parametry jako u příkladu vyhledávání.
- Do stromu postupně vložíme klíče 3, 22, 10, 2, 17, 5, 66, 68, 50, 7, 55, 45, 70, 44, 6, 21, 67, 1, 4, 8, 9, 12 a 15.

Příklad vkládání do B-stromu – vložení klíčů 3, 22 a 10

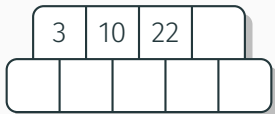
Vložení klíče 3



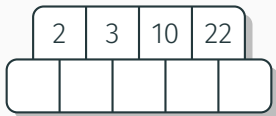
Vložení klíče 22



Vložení klíče 10

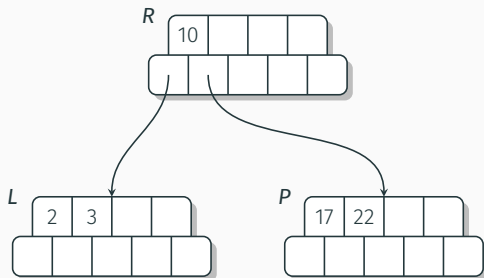


Příklad vkládání do B-stromu – vložení klíče 2



Stránka je zcela zaplněna, vložení libovolného dalšího klíče způsobí změnu struktury B-stromu.

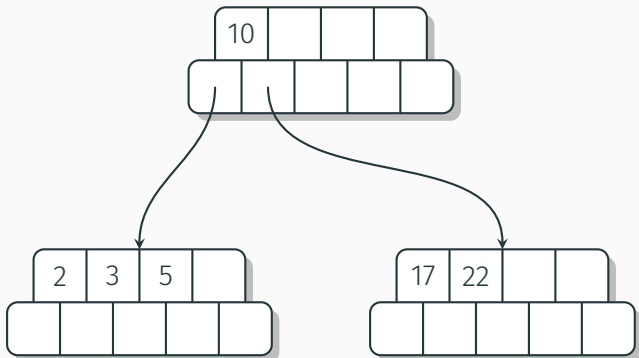
Příklad vkládání do B-stromu – vložení klíče 17



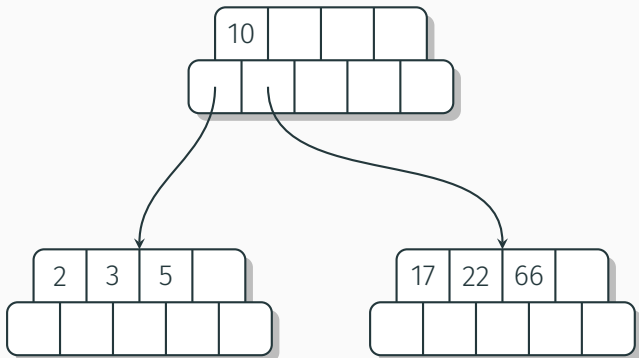
Vložení klíče 17 došlo:

1. k rozštěpení stránky **L** a k přesunu poloviny klíčů do nové stránky **P**,
2. ke vzniku nové kořenové stránky **R** a k přesunu klíče 10 do nového kořene.

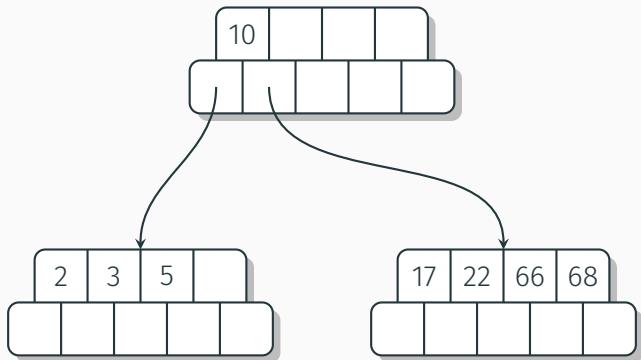
Příklad vkládání do B-stromu – vložení klíče 5



Příklad vkládání do B-stromu – vložení klíče 66

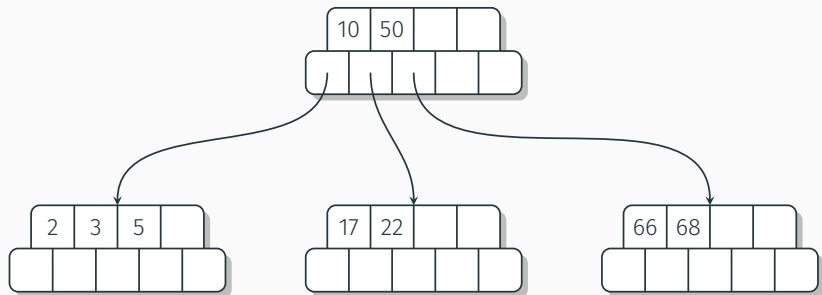


Příklad vkládání do B-stromu – vložení klíče 68



Stránka s klíči 17 až 68 je zcela zaplněna, vložení dalšího klíče do této stránky způsobí změnu struktury B-stromu.

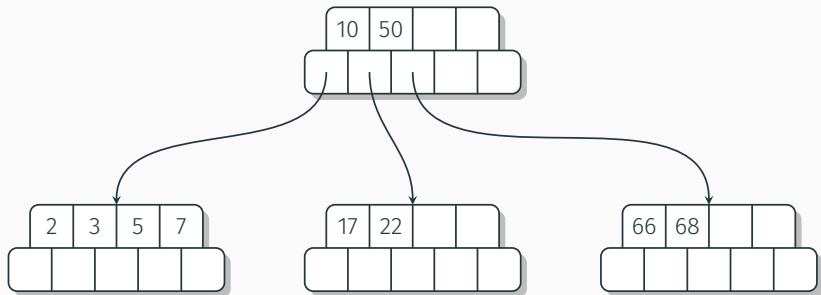
Příklad vkládání do B-stromu – vložení klíče 50



Vložení klíče 50 došlo:

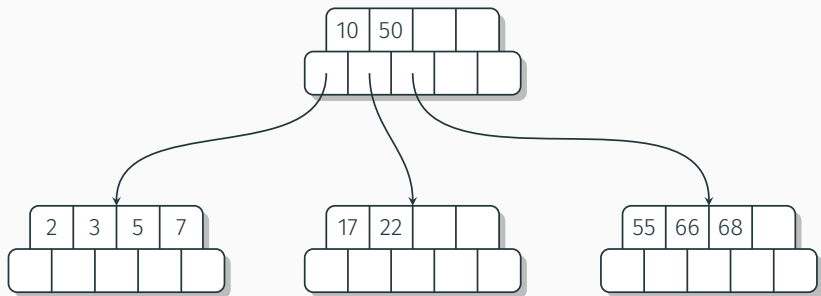
1. k rozštěpení stránky a k přesunu poloviny klíčů do nové stránky a
2. současně byl nově vložený klíč 50, jakožto medián hodnot v původní stránce, přesunut do kořenové stránky.

Příklad vkládání do B-stromu – vložení klíče 7

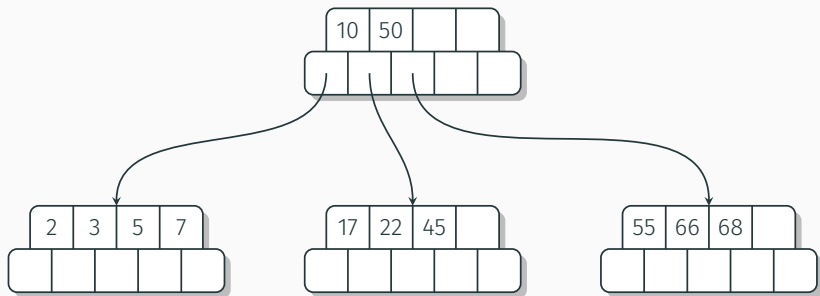


Stránka s klíči 2 až 7 je zcela zaplněna, vložení dalšího klíče do této stránky způsobí změnu struktury B-stromu.

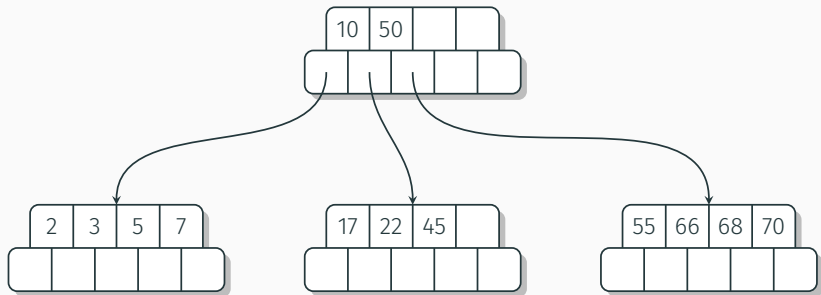
Příklad vkládání do B-stromu – vložení klíče 55



Příklad vkládání do B-stromu – vložení klíče 45

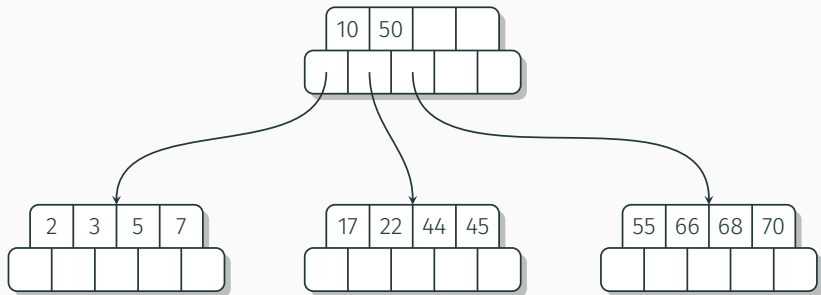


Příklad vkládání do B-stromu – vložení klíče 70



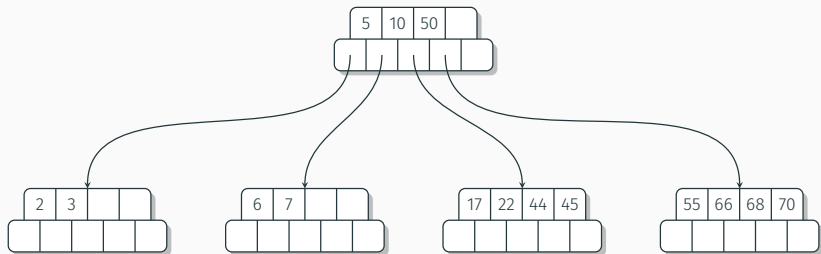
Stránka s klíči 55 až 70 je zcela zaplněna, vložení dalšího klíče do této stránky způsobí změnu struktury B-stromu.

Příklad vkládání do B-stromu – vložení klíče 44



Všechny listové stránky jsou zcela zaplněny, vložení jakéhokoliv dalšího klíče způsobí změnu struktury B-stromu.

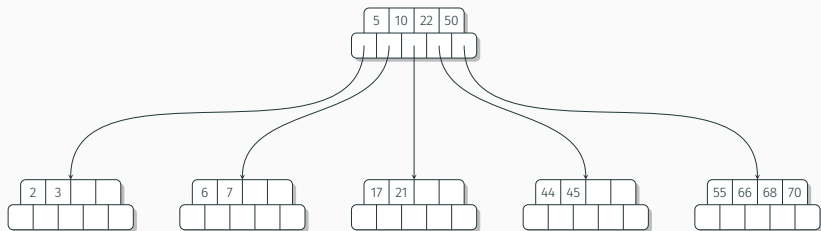
Příklad vkládání do B-stromu – vložení klíče 6



Vložením klíče 6 došlo:

1. k rozštěpení stránky a k přesunu poloviny klíčů do nové stránky a
2. současně k přesunu klíče 5 do kořenné stránky.

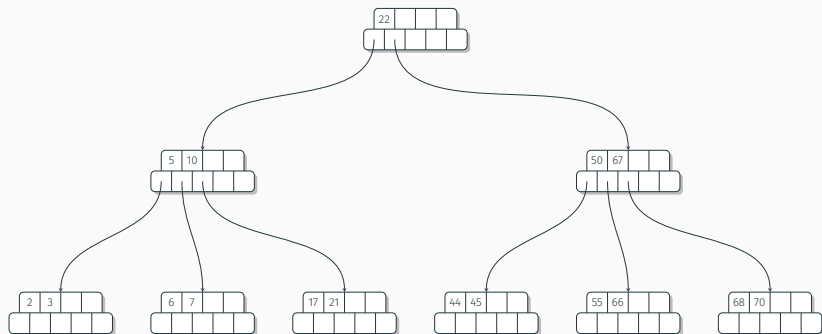
Příklad vkládání do B-stromu – vložení klíče 21



Vložení klíče 21 došlo:

1. k rozštěpení stránky a k přesunu poloviny klíčů do nové stránky a
2. k přesunu klíče 22 do rodičovské stránky.
3. Zároveň došlo k zaplnění kořenové stránky stromu.

Příklad vkládání do B-stromu – vložení klíče 67



Vložením klíče 67 došlo:

1. k rozštěpení stránky a k přesunu poloviny klíčů do nové stránky a
2. současně byl nově vložený klíč 67, jakožto medián hodnot v původní stránce, přesunut do rodičovské stránky.
3. Protože i tato stránka byla již zcela zaplněna, došlo k jejímu rozštěpení a tím pádem k vytvoření nové kořenové stránky stromu s jediným klíčem 22.

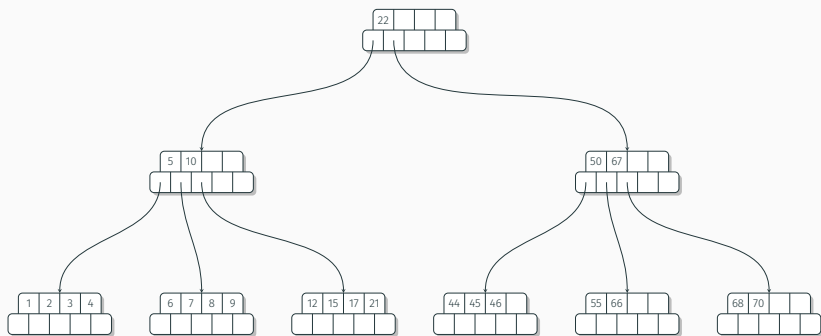
Poznámky

- V tomto okamžiku dosahuje faktor naplnění B-stromu minimální hodnoty $\approx 50\%$. V B-stromu je maximum volného místa pro vkládání dalších klíčů.
- Minimální zaplnění B-stromu ale naopak způsobí, že odebráním libovolného klíče dojde ke slučování stránek¹ B-stromu, včetně zrušení kořene a následného snížení výšky B-stromu.

¹Viz další část prezentace

Příklad vkládání do B-stromu – vložení dalších klíčů

Do stromu byly dále vloženy klíče 1, 4, 8, 9, 12, 15 a 46. Na pořadí vložení klíčů, v tomto případě, nezáleží.



1. Nejprve je nutné klíč x ve stromu najít.
2. Označme stránku s klíčem x jako P .
3. Mohou nastat dva případy:
 - stránka P je **vnitřní stránka** stromu nebo
 - stránka P je **listová stránka**.

B-stromy – smazání klíče x z vnitřní stránky P

1. Klíč x ve stránce P nahradíme k němu nejbližším větším klíčem y .
2. Klíč y se musí nacházet v podstromu s klíči většími než x a zároveň je z těchto klíčů nejmenší, musí se tedy nacházet v listové stránce.
3. Smazání klíče x z vnitřní stránky stromu jsme tak převedli na smazání klíče y z listové stránky stromu.

B-stromy – smazání klíče x z listové stránky P

1. Klíč x vymažeme ze stránky P .
2. Pokud stránka P i potom obsahuje aspoň n klíčů, je proces smazání ukončen.
3. Pokud P potom obsahuje jen $n - 1$ klíčů, musíme chybějící jeden klíč doplnit.
 - 3.1 Zjistíme počet klíčů v sourozenci stránky P . Sourozence označíme jako S . Společného rodiče stránek P a S označíme jako R .
 - 3.2 Pokud je v S je **více než n** klíčů, potom
 - 3.2.1 nejbližší větší klíč než x přesuneme z R do P a
 - 3.2.2 nejmenší z klíčů v S přesuneme do R .
 - 3.3 Ve stránce S je **přesně n** klíčů, potom

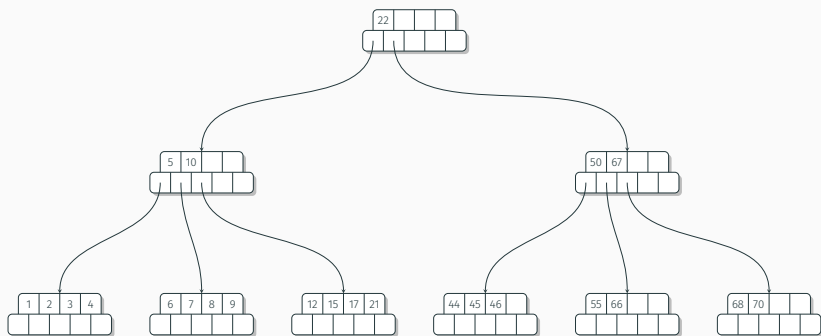
- 3.3.1 klíče ze stránky S přesuneme do stránky P a dostáváme jednu stránku s $2n - 1$ klíči.
- 3.3.2 Stránku S zrušíme.
- 3.3.3 Ve stránce R je nyní jeden odkaz na stránku přebytečný. Nejbližší větší klíč než x přesuneme z R do stránky P , která nyní obsahuje přesně $2n$ klíčů.

B-stromy – smazání klíče x z listové stránky P (pokrač.)

Poznámky

- Obvykle volíme sourozence s většími klíči než je x , tedy sourozence „vpravo“ od P . V předchozím výkladu jsme předpokládali tuto volbu.
- Lze však volit i sourozence s menšími klíči, čili „vlevo“ od P . Další postup je v tomto případě zrcadlovým obrazem k sourozenci „vpravo“.
- Proces přesunu klíčů z S do P a následné zrušení stránky S se nazývá **slučování stránek**.
- Proces slučování stránek může pokračovat postupně až ke kořeni stromu a může vést i k zániku současného kořene stromu. Novým kořenem stromu pak bude stránka vzešlá z procesu slučování. B-strom tak sníží svoji výšku.

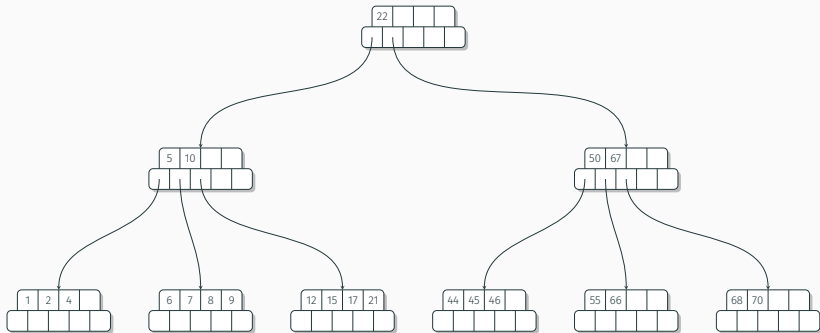
Příklad mazání v B-stromu – výchozí B-strom



V tomto stavu jsme B-strom zanechali na konci příkladu vkládání klíčů do B-stromu. Nyní budeme klíče postupně z B-stromu mazat.

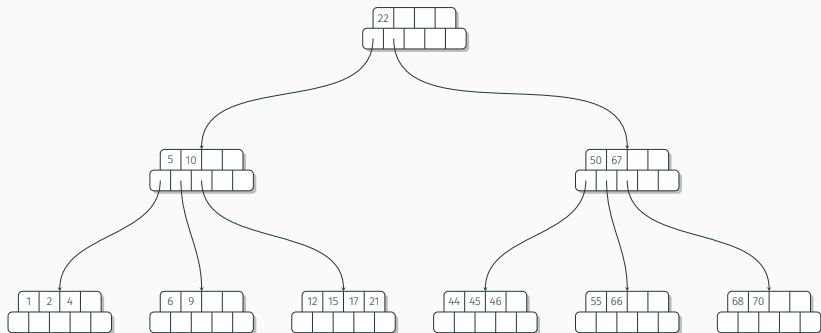
Příklad mazání v B-stromu – smazání klíče 3

Klíč 3 se nachází v listové stránce, kde je dostatečný počet klíčů na to, aby chom klíč 3 mohli jednoduše smazat. Tím dostáváme následující B-strom.



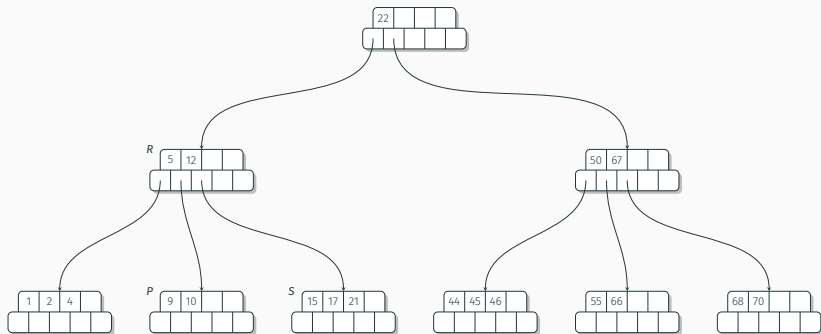
Příklad mazání v B-stromu – smazání klíčů 7 a 8

Stejným způsobem smažeme klíče 7, 8 a dostáváme



Příklad mazání v B-stromu – smazání klíče 6

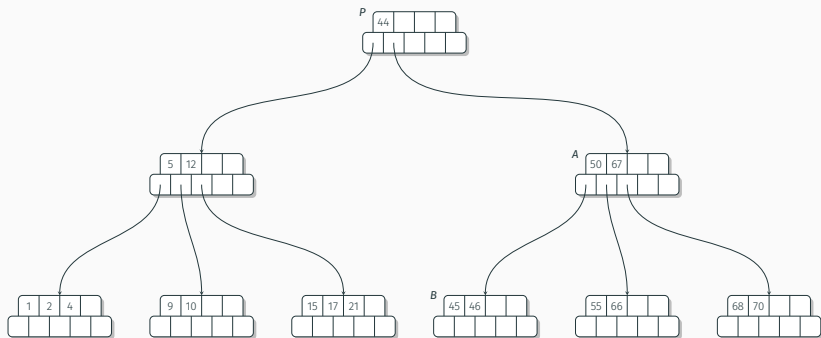
1. Po smazání klíče 6 obsahuje stránka P $n - 1 = 1$ klíč, číslo 9.
2. Sourozenec S obsahuje více než n klíčů.
3. Nejbližší větší klíč než 6, tj. 10, přesuneme z R do P .
4. Nejmenší klíč z S , tj. 12, přesuneme do R .



Příklad mazání v B-stromu – smazání klíče 22

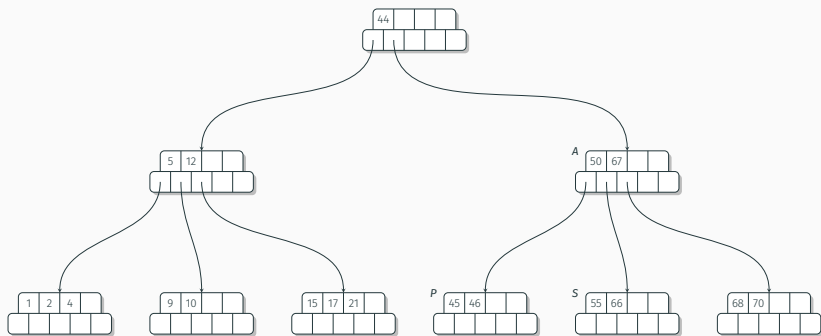
1. Klíč 22 se nachází ve vnitřní stránce **P**, viz následující obrázek.
2. Nahradíme jej nejbližším větším klíčem – větší klíče než 22 jsou v podstromu s kořenem ve stránce **A**. Odtud pokračujeme do nejlevější listové stránky, v našem případě do **B**.
3. Vybereme nejmenší klíč v **B**, tj. 44.
4. Tím jsme převedli smazání klíče 22 na smazání klíče 44.
5. Po provedení všech operací, které odpovídají smazání 44 (v tomto případě se jedná jen o smazání 44 ze stránky **B**), nahradíme klíčem 44 klíč 22.

Příklad mazání v B-stromu – smazání klíče 22 (pokrač.)



Příklad mazání v B-stromu – smazání klíče 46, fáze I

Stav B-stromu před zahájením mazání

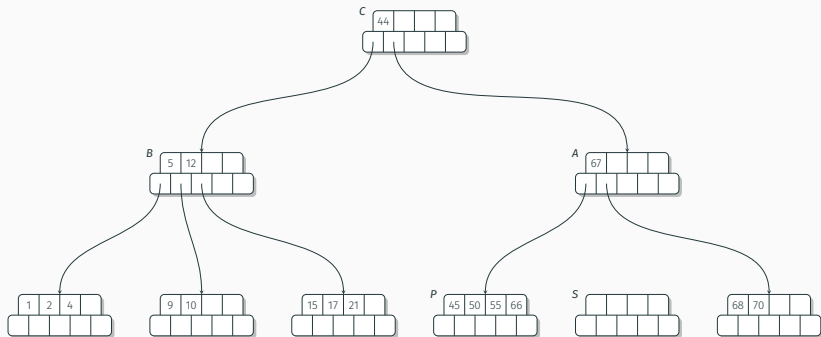


Příklad mazání v B-stromu – smazání klíče 46, fáze I (pokrač.)

1. Po smazání klíče 46 obsahuje stránka P $n - 1$ klíčů, tj. pouze klíč 45.
2. Sourozenec S obsahuje přesně než n klíčů, musíme slučovat stránky.
3. Všechny klíče z S přesuneme do P .
4. Stránka P je prvním potomkem stránky A , přesuneme tedy i první klíč z A do P .

Příklad mazání v B-stromu – smazání klíče 46, fáze I (pokrač.)

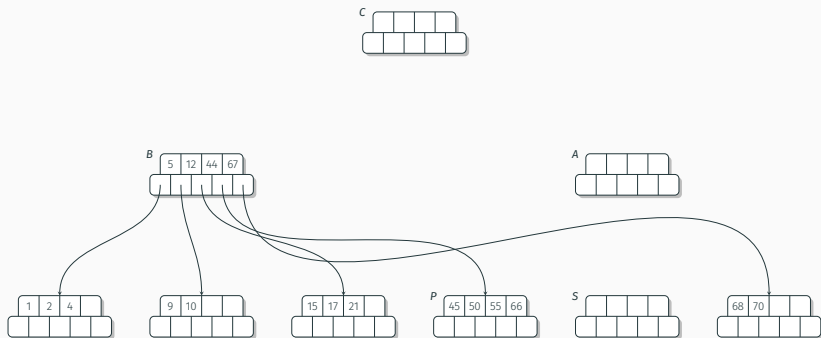
Výsledek fáze 1



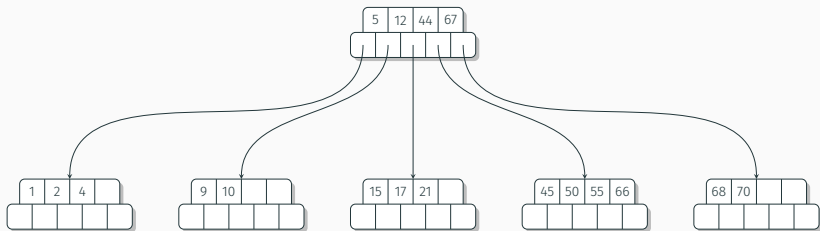
Příklad mazání v B-stromu – smazání klíče 46, fáze II

1. Ve stránce **A** zůstalo jen $n - 1$ klíčů, což odporuje definici B-stromu.
2. Sourozenec **B** obsahuje přesně n klíčů, musíme i na této úrovni provést slučování stránek.
3. Přesuneme klíč 67 ze stránky **A** do stránky **B**.
4. A stejně tak do **B** přesuneme i jeden klíč z rodičovské stránky **C**.
5. Tím došlo ke zrušení kořenové stránky a ke snížení výšky B-stromu.

Příklad mazání v B-stromu – smazání klíče 46, fáze II (pokrač.)

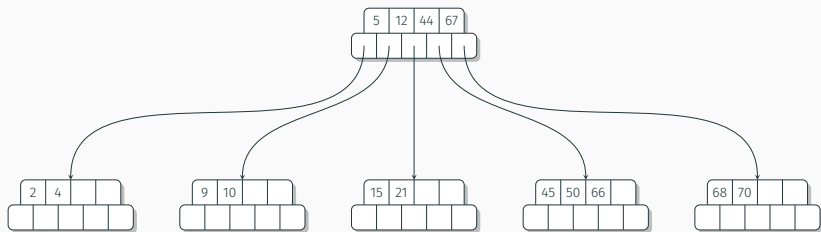


Příklad mazání v B-stromu – smazání klíče 46, výsledek



Příklad mazání v B-stromu – smazání klíčů 1, 17 a 55

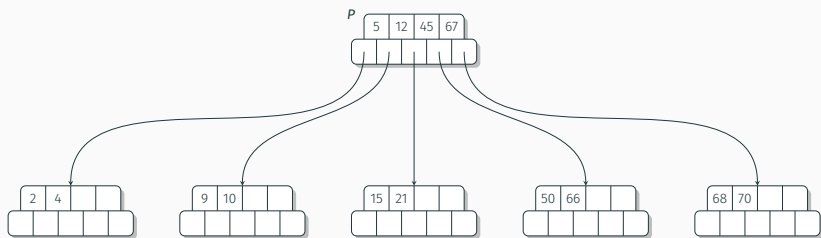
Klíče 1, 17 a 55 se nachází v listových stránkách, kde je dostatečný počet klíčů na to, abychom je mohli jednoduše smazat.



Příklad mazání v B-stromu – smazání klíče 44

1. Klíč 44 se nachází ve vnitřní stránce.
2. Nahradíme jej nejbližším větším klíčem, tj. klíčem 45.

Výsledný strom

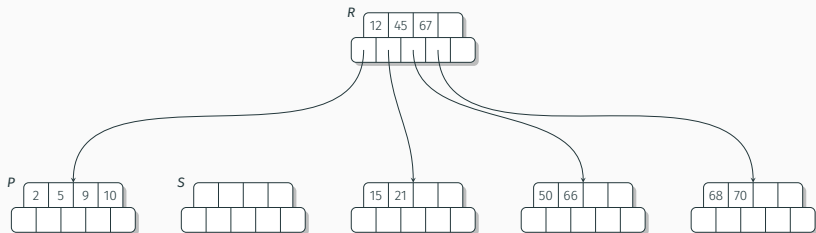


B-strom se nyní nachází ve stavu, kdy listové stránky jsou naplněny na minimální přípustnou úroveň. Smazání jakéhokoliv klíče způsobí sloučení stránek.

Příklad mazání v B-stromu – smazání klíče 4

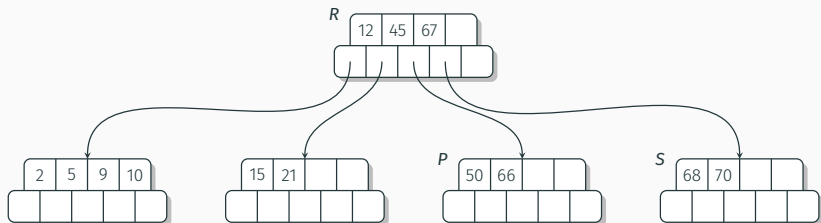
1. Po smazání klíče 4 obsahuje stránka P pouze $n - 1$ klíčů.
2. Sourozenec S obsahuje přesně n klíčů.
3. Klíče z S přesuneme do P .
4. Do P přesuneme i klíč 5 z rodičovské stránky R , protože v R by jinak jeden odkaz na potomky přebýval.

Výsledný strom



Příklad mazání v B-stromu – smazání klíče 45

Strom před smazáním klíče 45



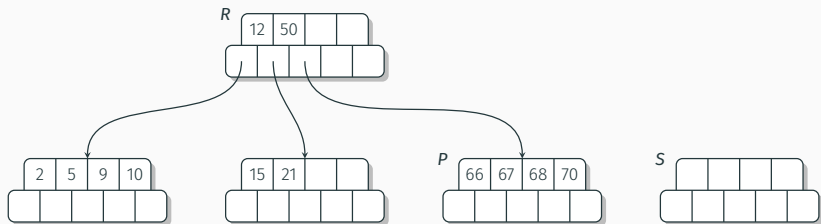
1. Klíč 45 se nachází ve vnitřní stránce *R*.
2. Nahradíme jej nejbližším větším klíčem, tj. klíčem 50.
3. Tím jsme převedli smazání klíče 45 na smazání klíče 50.

Příklad mazání v B-stromu – smazání klíče 45 (pokrač.)

4. Po smazání klíče 50 obsahuje stránka P pouze $n - 1$ klíčů.
5. Sourozenec S obsahuje přesně n klíčů.
6. Klíče z S přesuneme do P .
7. Do P přesuneme i nejbližší větší klíč než 45, tj. klíč 57, z rodičovské stránky R , protože v R by jinak jeden odkaz na potomky přebýval.

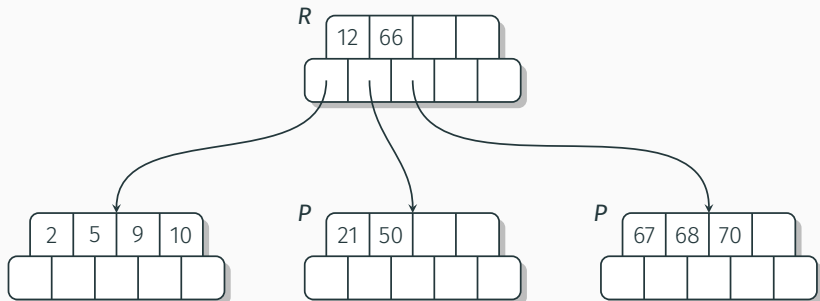
Příklad mazání v B-stromu – smazání klíče 45 (pokrač.)

Výsledný strom



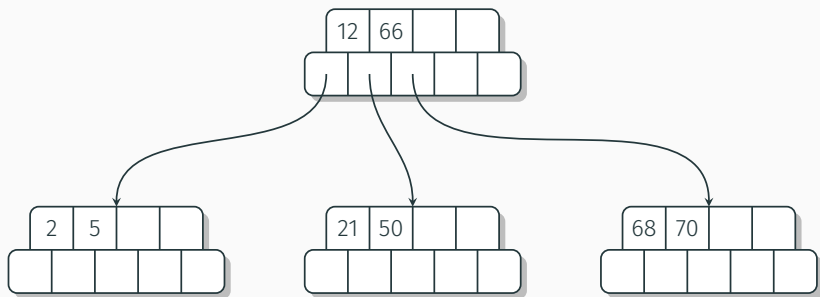
Příklad mazání v B-stromu – smazání klíče 15

Po smazání 15 zůstalo ve stránce *P* pouze $n - 1$ klíčů. Musíme tedy přesunout přes stránku *R* jeden klíč ze stránky *S*.



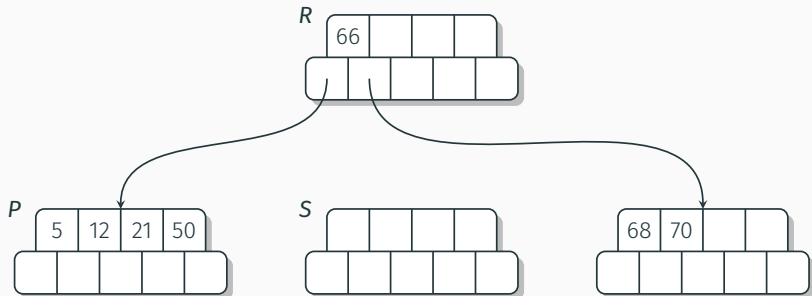
Příklad mazání v B-stromu – smazání klíčů 9, 10 a 67

Smazání klíčů 9, 10 a 67 je velice jednoduché.

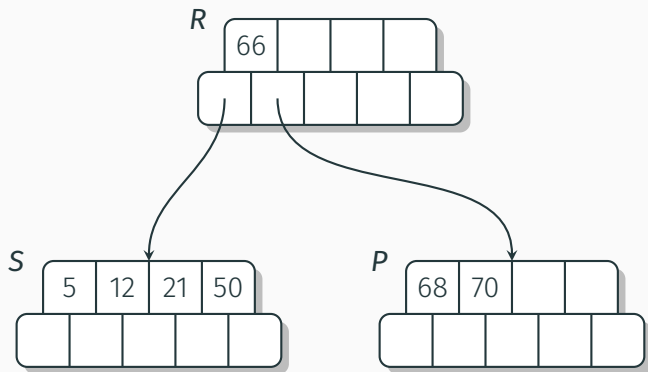


Příklad mazání v B-stromu – smazání klíče 2

Po smazání 2 zůstalo ve stránce P pouze $n - 1$ klíčů. Sourozenec S obsahuje n klíčů, dojde tedy ke slučování stránek.



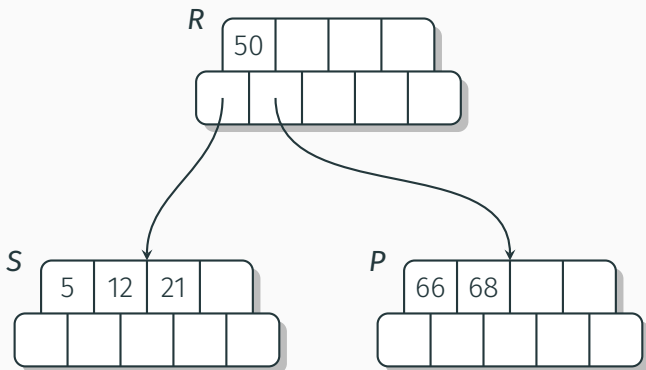
Příklad mazání v B-stromu – smazání klíče 70



Příklad mazání v B-stromu – smazání klíče 70 (pokrač.)

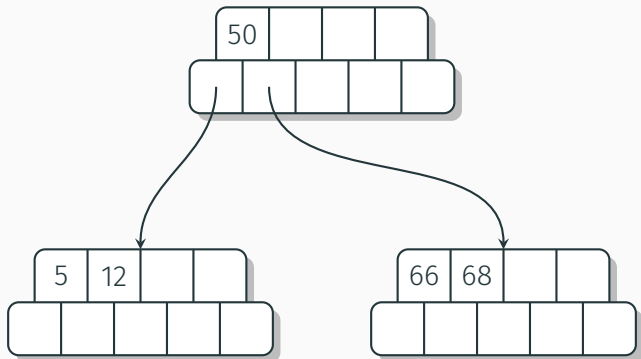
Po smazání 70 zůstalo ve stránce P pouze $n - 1$ klíčů.

Sourozenec S obsahuje více n klíčů, dojde tedy k přesunu 66 z R do P a nejbližšího menšího klíče z S do R .



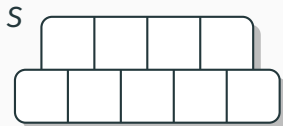
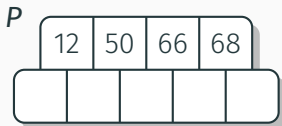
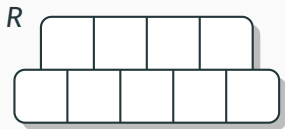
Příklad mazání v B-stromu – smazání klíče 21

Smazání klíče 21 je velice jednoduché.



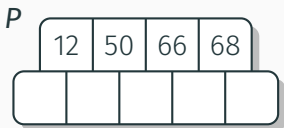
Příklad mazání v B-stromu – smazání klíče 5

Smazání klíče 5 je zřejmé.



Příklad mazání v B-stromu – smazání klíče 5 (pokrač.)

Stránka P se stala novým kořenem, a současně jedinou stránkou, B-stromu.



Smazání klíčů 12, 50, 66 a 68 je už triviální záležitostí.

Děkuji za pozornost