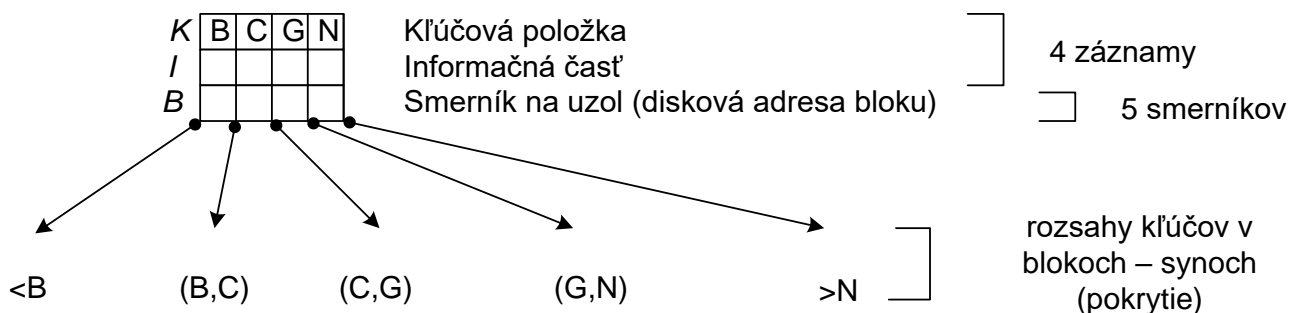


# B - strom

- viaccestné zovšeobecnenie binárneho vyhľadávacieho stromu a 2-3 stromu
- využívajú sa princípy fungovania 2-3 stromu, ale dáta sú uložené na disku
- vrchol stromu je nahradený jediným blokom dát na disku, namiesto smerníkov sa používajú diskové adresy

## Organizácia:

### Štruktúra uzla stromu (bloku na disku):

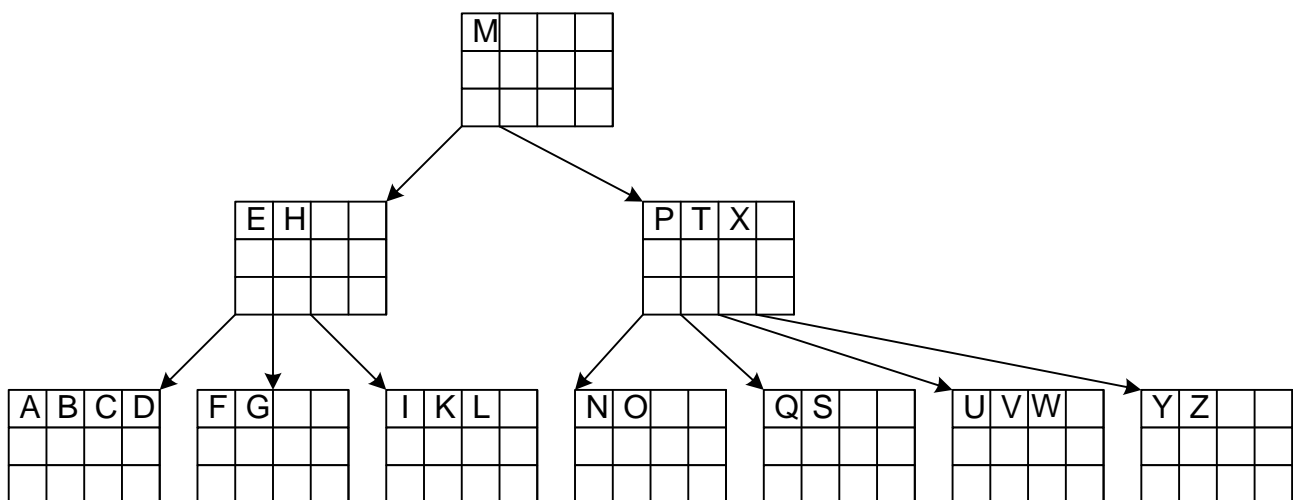


Maximálny počet  $m$  synov uzla sa nazýva rád B- stromu.

Maximálny počet záznamov v uzle takéhoto bloku je  $m - 1$ . Každý vnútorný uzol B - stromu rádu  $m$  musí mať najmenej  $k = \lceil m/2 \rceil - 1$  záznamov (okrem koreňa, ten musí mať najmenej 1 záznam).

Uzol ukázaný na obrázku je uzlom B - stromu rádu 5. Obsahuje 2 až 4 záznamy a ukazuje na 3 až 5 synov.

### Príklad B- stromu rádu 5:



### **Ďalšie vlastnosti:**

- všetky listy sú na rovnakej úrovni (2-3 strom je dokonale vyvážený strom)
- výška maximálne zaplneného B- stromu je  $\log_m b$  (b je počet blokov)
- rád stromu býva v praxi veľký
- záznamy v blokoch sú utriedené

### **Spracovanie:**

**Teoreticky aj sekvenčné v poradí kľúčov (aplikáciou prehľadávania inorder).**

### **Operácia nájsť záznam s kľúčom K:**

- sprístupni koreň
- v aktuálnom bloku nájsť buď záznam s kľúčom K, alebo adresu bloku, ktorý ho pokrýva
- sprístupni pokrývajúci blok a opakuj (v najhoršom prípade) po sprístupnenie a prehliadku listu

**Počet blokových prenosov v najhoršom prípade rovný výške B – stromu.**

### **Operácia vlož:**

1. Vyhľadaj záznam s kľúčom K. Ak sa nájde, hlás chybu duplicity, inak aktuálnym blokom A je blok do ktorého záznam patrí.
2. Ak v aktuálnom bloku A je miesto, vlož doňho záznam s prípadným preusporiadaním záznamov v bloku kvôli usporiadanosti. Koniec vkladania.
3. Ak je aktuálny blok zaplnený:
  - alokuj v súbore nový blok B,
  - z kľúčov záznamov aktuálneho bloku a kľúča K nájsť medián M,
  - záznamy s kľúčom väčším ako medián zapíš do bloku B, menšie ako medián ponechaj v aktuálnom,
  - choď do otca doteraz aktuálneho bloku a pokús sa vložiť záznam s kľúčom M (choď na krok 2).
  - V prípade, že sa príde až do koreňa, tak sa výška stromu zvýši o jednu úroveň.

Ak je plný aj koreň, vzrastie výška stromu o 1 (alokuje sa nový koreň, do ktorého sa ako jediný vloží záznam s kľúčom rovným mediánu).

Voľné miesta v blokoch slúžia ako “vata”, aby sa často nevykonávala reorganizácia.

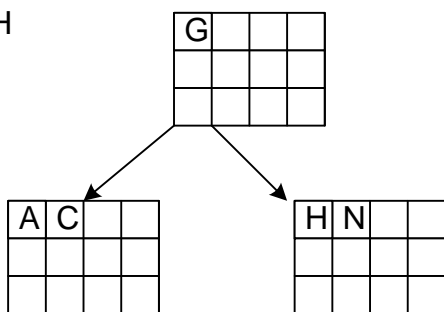
### Príklad:

Vyjdeme z prázdneho B- stromu rádu 5 a budeme doňho postupne vkladať záznamy.

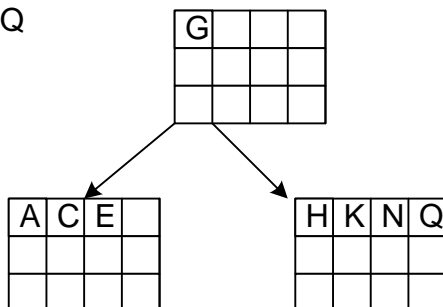
Po postupnom vložení C, N, G, A

A	C	G	N

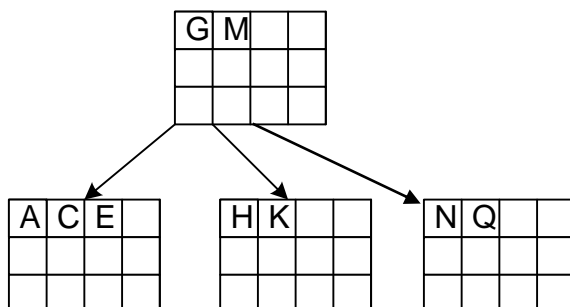
Po vložení H



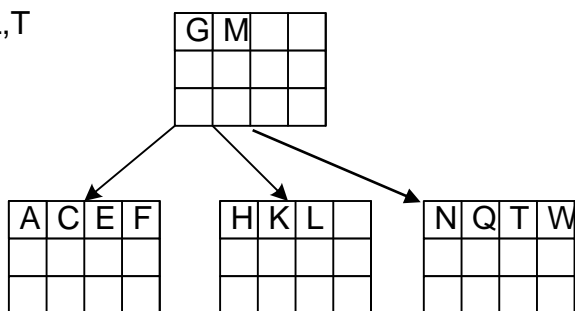
Po vložení E, K, Q



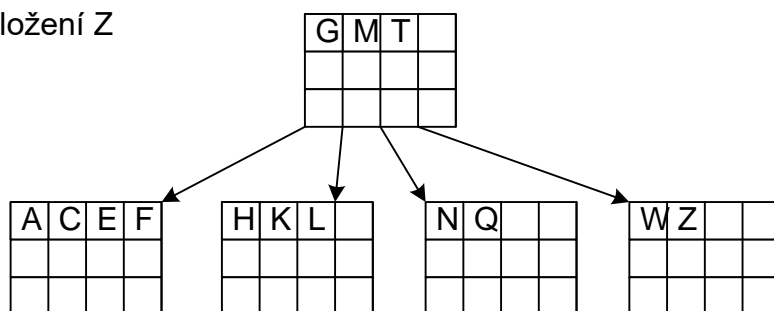
Po vložení M



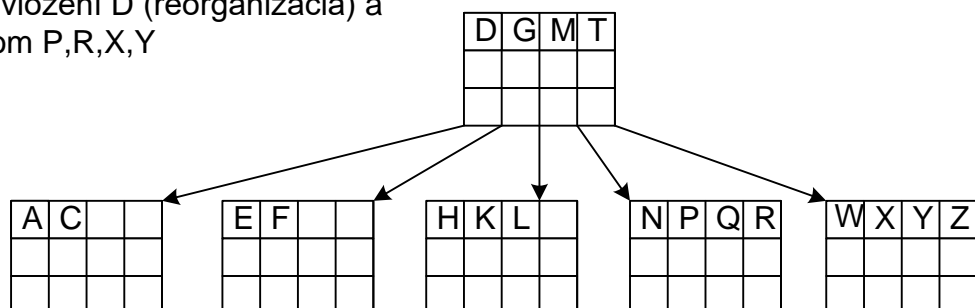
Po vložení F,W,L,T



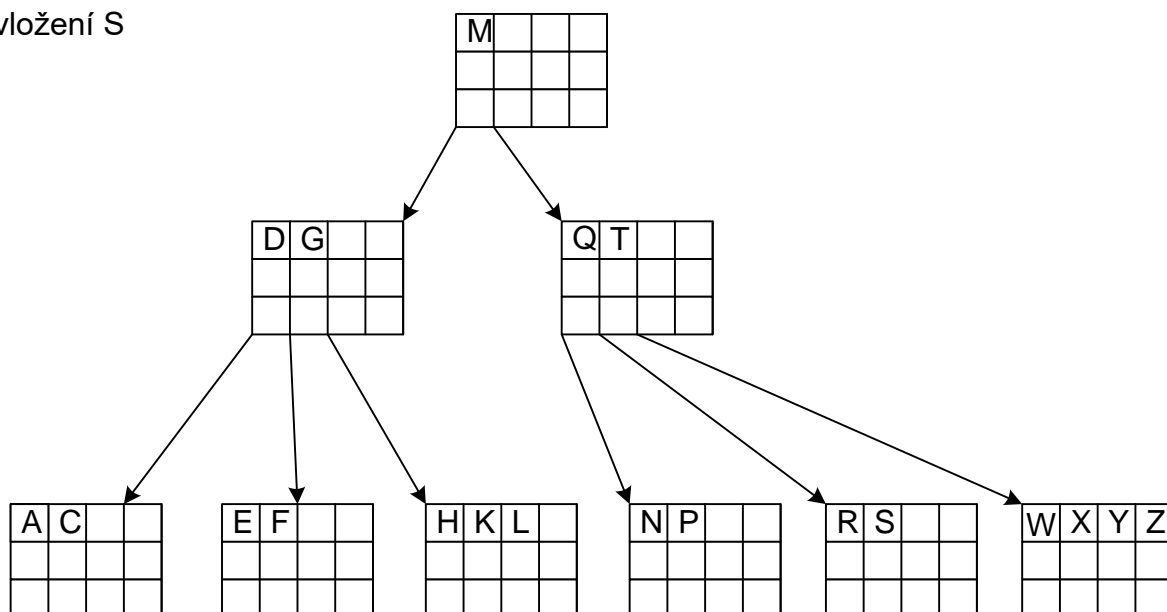
Po vložení Z



Po vložení D (reorganizácia) a potom P,R,X,Y



Po vložení S

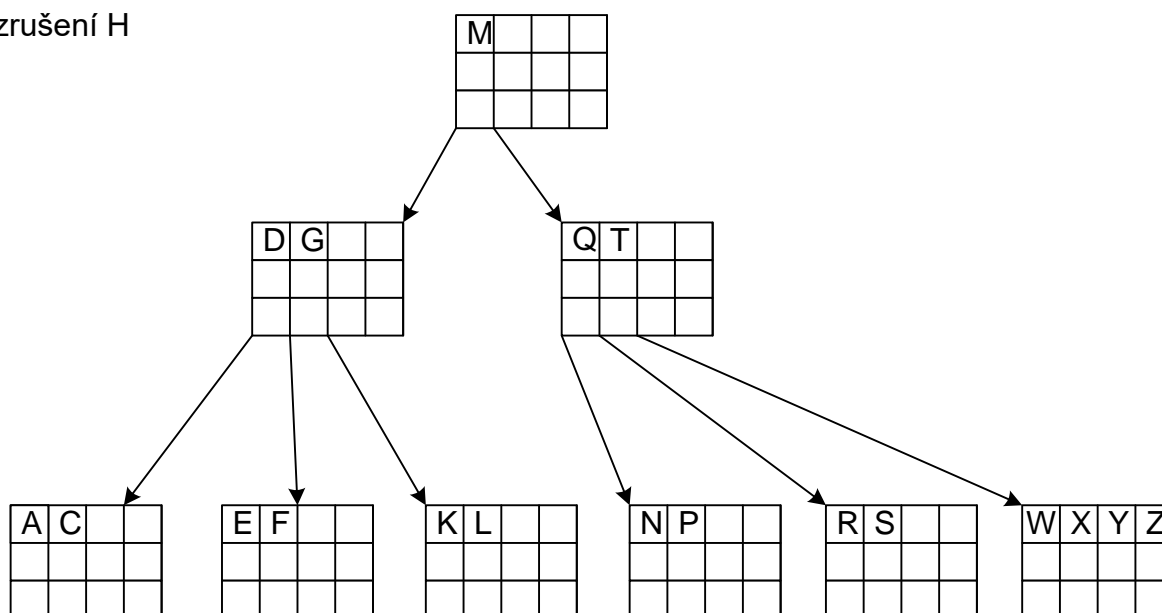


## Operácia vymaž:

Nájdí záznam s kľúčom K. A je aktuálny blok, P je jeho otec.  
Môže nastať niekoľko prípadov, ktorých riešenie ukážeme iba na príklade.

Zrušme v naposledy uvedenom B- strome záznam s kľúčom H.  
Záznam je v bloku-liste a po zrušení neklesne v ňom počet záznamov pod minimálny počet  $k$ . Riešenie: jednoduché zrušenie bez reorganizácie:

Po zrušení H

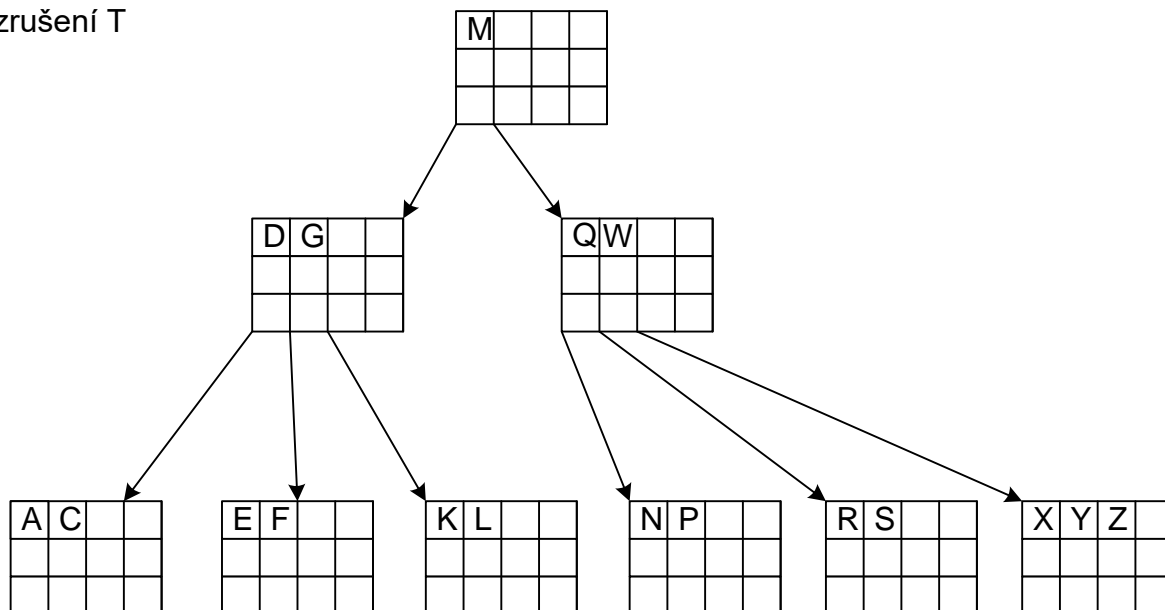


Teraz zrušme záznam s kľúčom T.

Blok A v ktorom je T nie je listom.

Riešenie: Záznam s kľúčom T nahradíme záznamom s kľúčom, ktorý je nasledovníkom rušeného kľúča T podľa veľkosti (je to záznam s kľúčom W). Záznam s kľúčom W v pôvodnom bloku zrušíme.

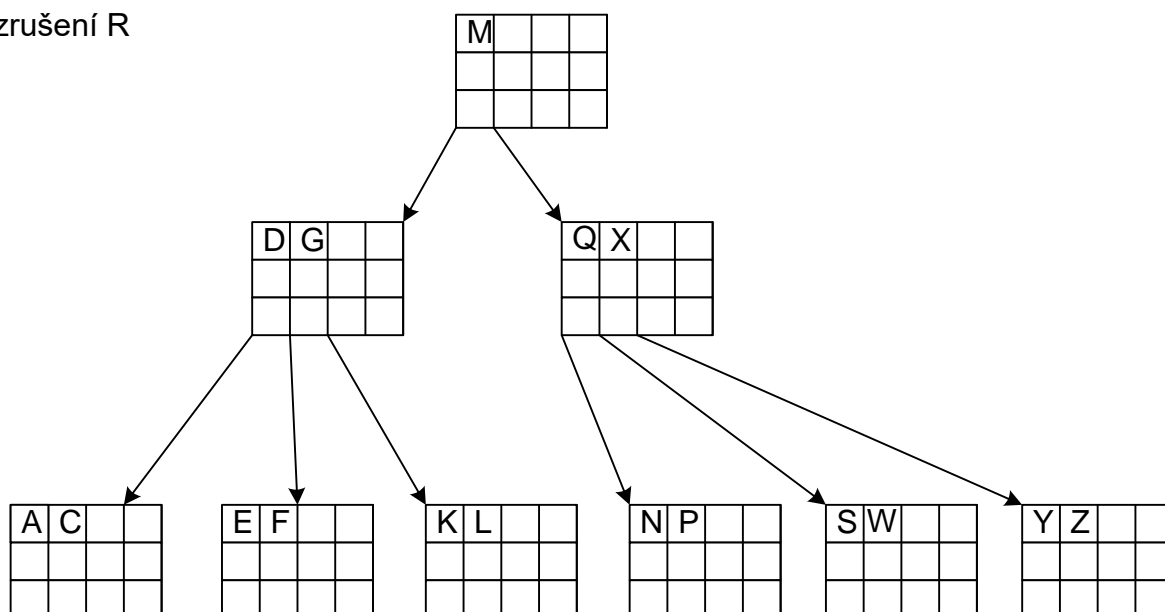
Po zrušení T



**Zrušíme záznam s kľúčom R. Tento záznam je v bloku-liste A, ktorý má minimálny povolený počet záznamov  $k$  a po jeho zrušení by sa obsah bloku stal nekorektným.**

**Riešenie:** Zistíme, či jeden z besprostredných bratov A má viac ako  $k$  záznamov (u nás je to pravý brat). Prvý záznam (s kľúčom X) z tohoto bloku vyberieme a nahradíme ním záznam ukazujúci na tento blok v jeho otcovi (u nás záznam s kľúčom W). “Vytesnený” záznam (s kľúčom W) vložíme do aktuálneho bloku A, v ktorom súčasne vymažeme záznam s kľúčom R.

Po zrušení R

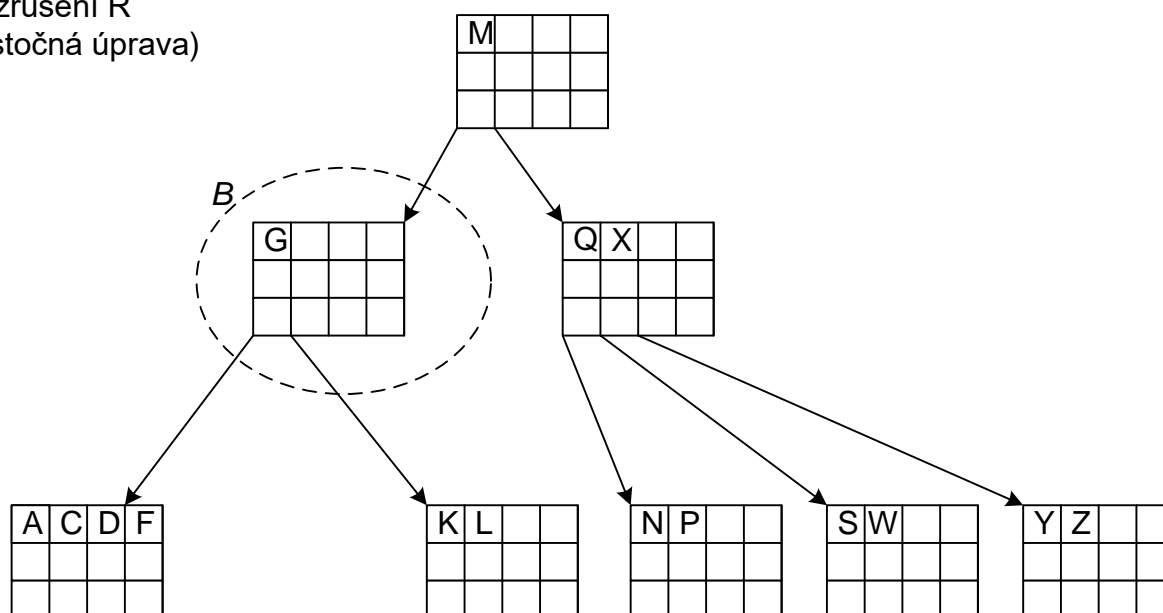


**Zrušenie záznamu s kľúčom E.**

**Záznam je v bloku – liste (označme ho A). Obidvaja bezprostrední bratia majú minimálny počet záznamov  $k$  a preto nie je možné od nich záznam “požičiať”.**

**Riešenie:** Spojíme obsah bloku A a jeho (napríklad) ľavého brata a jeden blok dealokujeme. Záznam s kľúčom D, ktorý doteraz “rozdeľoval” dva spojené bloky odoberieme z jeho súčasnej pozície a vložíme ho do spojeného bloku. Výsledkom je nasledujúca štruktúra, ktorá však zatiaľ nie je korektná, pretože jeden blok ( $B$ ) má nepovolený počet záznamov.

Po zrušení R  
(čiastočná úprava)

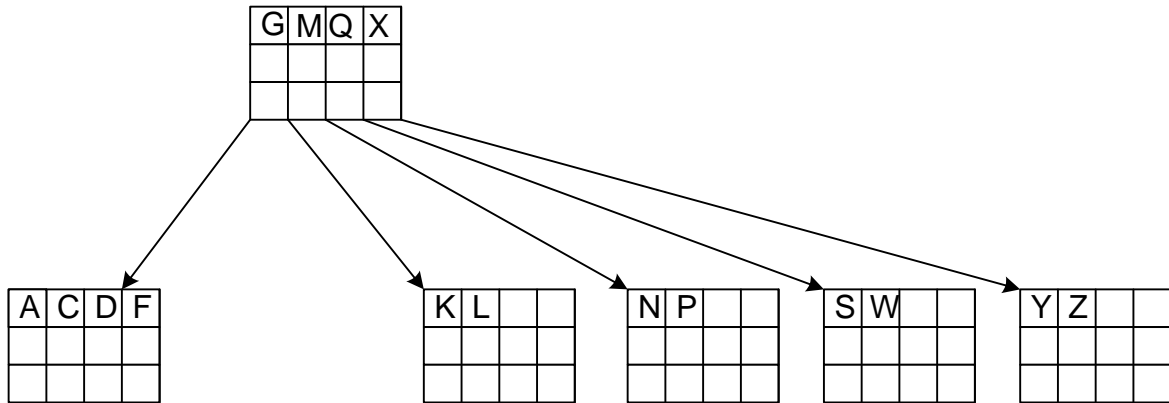


**Ďalšia úprava má charakter cyklického opakovania:**

**Ak by blok B mal bezprostredného brata s počtom záznamov  $> k$ , (napr. vpravo), blok  $B$  by si od neho „požičal“ záznam známym spôsobom: záznam s kľúčom M by sa premiestnil do bloku  $B$  a na jeho miesto by sa presunul záznam s kľúčom Q. Okrem toho by sa ľavý podstrom Q stal pravým podstromom M.**

**V našom prípade nie je možné „požičať“. Preto sa spojí blok A s jeho pravým bratom a spolu s jediným záznamom svojho otca vytvoria blok so záznamami (G, M, Q, X). Jeden z dvojice blokov - bratov dealokujeme. Keďže sa však doterajší koreň vyprázdnil, dealokujeme ho tiež, čím B-strom zníži svoju výšku.**

Po zrušení R



**Štruktúra je používaná v súborových systémoch (HFS+ (Apple), btrfs (B-tree file system) and Ext4).**