## B+ strom

**B+ strom** je stromová datová struktura vycházející z [B-stromu](http://referaty-seminarky.cz/b-strom/) umožňující rychlé vkládání, vyhledávání i mazání dat. Data jsou zpřístupněna pomocí klíčů, přičemž na rozdíl od [B-stromu](http://referaty-seminarky.cz/b-strom/) jsou **všechna data**uložena až na samém konci stromu, v listech. Klíče jsou však uloženy i ve vnitřních uzlech či v kořenu. Ostatní vlastnosti má stejné jako zmiňovaný [B-strom](http://referaty-seminarky.cz/b-strom/).

B+ strom poprvé popsal[Rudolf Bayer](http://referaty-seminarky.cz/rudolf-bayer/) spolu s Edwardem M. McCreightem roku [1972](http://referaty-seminarky.cz/1972/) v „Organization and Maintenance of Large Ordered Indices. Acta Informatica 1: 173-189 (1972)“.

## Vlastnosti B+ stromu

* Všechny listy (tj.uzly které nemají žádné potomky) jsou na stejné úrovní (ve stejné hloubce).
* Data můžou být uložena **pouze** v listech
* Všechny uzly kromě kořene mají maximálně *N* a minimálně *N/2* potomků (přesněji:left lceil frac{n}{2} 
  ight 
  ceil -1 ).
* Kořen má nejvýše *N* potomků, spodní hranice není omezena.

### Charakteristické vlastnosti B+ stromu

Mějme B+ strom řádu *B* kde vzdálenost od kořene k listům je *H*:

* Maximální počet uložených záznamů je *N* = *BH*
* Minimální počet klíčů je 2(*B* / 2)*H*- 1
* Místo požadované pro uložení stromu je *O*(*N*)
* Vložení záznamu do stromu vyžaduje v nejhorším případě *O*(log*BN*) operací
* Vyhledání záznamu v nejhorším případě vyžaduje *O*(log*BN*) operací
* Vymazání (dříve nalezeného) záznamu v nejhorším případě vyžaduje *O*(log*BN*) operací
* Vyhledání více položek v rámci zadaného rozsahu trvá v nejhorším případe *O*(log*BN* + *K*)operací (*K* je zde počet položek vyskytujících se v dotazovaném rozsahu)

## Skutečná implementace B+ stromu

Skutečný B+ strom se ve skutečnosti realizuje tak, že je vždy ve všech listech uložen kromě vlastních klíčů a hodnot také odkaz (ukazatel) na následujícího sourozence. Díky tomu je umožněna rychlejší práce s bloky souvislých dat a s dotazy pracujícími s rozsahy typu *vrať všechny záznamy kde plat je mezi 10000-20000*. Tento jeden ukazatel navíc v rámci každého listu nijak dramaticky nezvětšuje paměťovou náročnost na uložení stromu, ale dramaticky zvyšuje výkon např. ve zmiňovaných souborových systémech. Tento mechanismus odkazů na následujícího sourozence je zobrazen i na obrázku červenými políčky.

## Použití B+ stromu

**B+ strom** je dynamická struktura, která je navržena pro práci s většími bloky dat (běžně je využíván pro ukládání dat na disk v blocích). Jeho výhodou je velice rychlé získávání (čtení) souvislého bloku dat (ve srovnání s ostatními druhy stromů). Pokud máme bloky dat definované délky *B*, do **B+ stromu** se uloží počet klíčů který je roven násobku *B* a čtení celého bloku dat je velice efektivní oproti [binárnímu vyhledávacímu stromu](http://referaty-seminarky.cz/binarni-vyhledavaci-strom/) (což je odpovídající blokově neorientovaná alternativa).

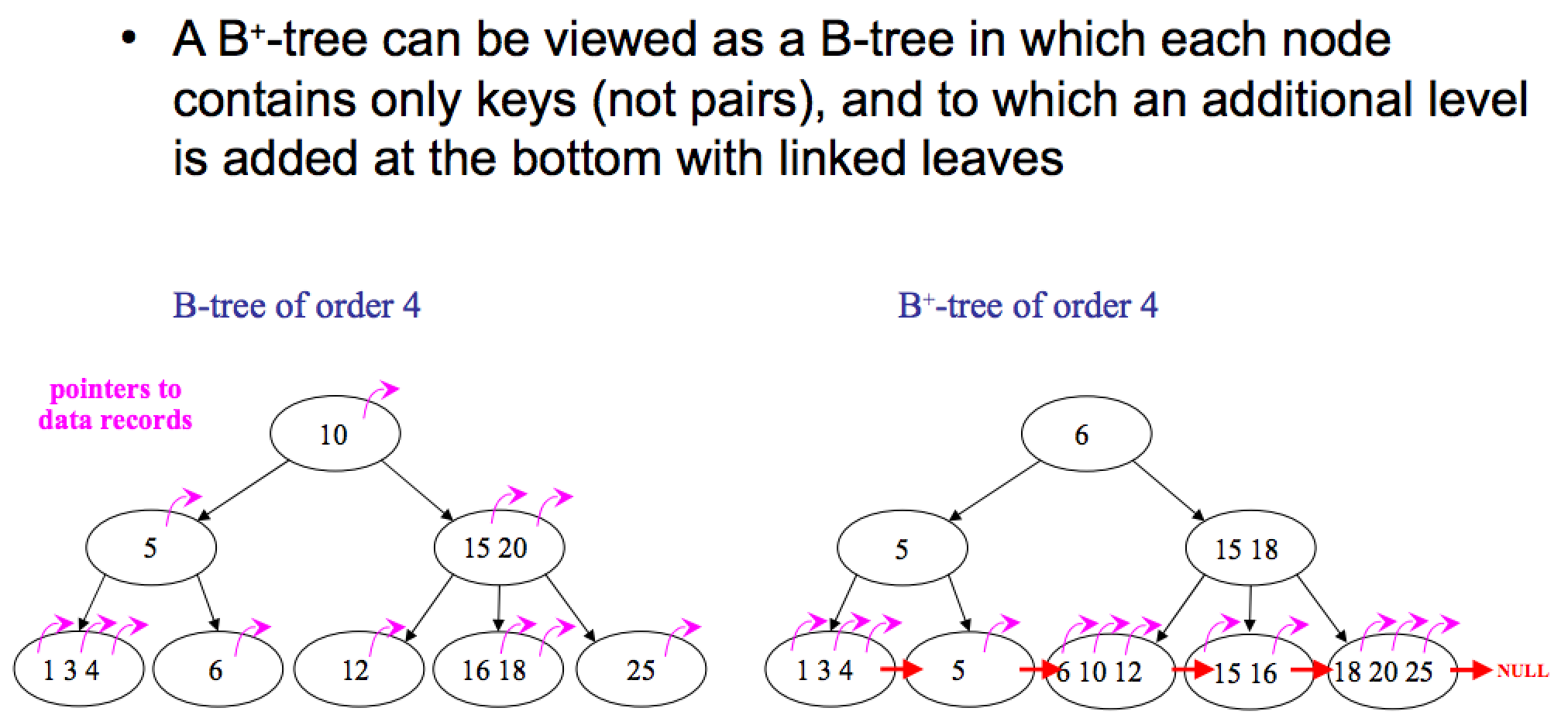
Tento systém používají pro indexování dat na disku [souborové systémy](http://referaty-seminarky.cz/souborovy-system/) [NTFS](http://referaty-seminarky.cz/ntfs/), [ReiserFS](http://referaty-seminarky.cz/reiserfs/), [XFS](http://referaty-seminarky.cz/xfs/) a JFS2. [Relační databáze](http://referaty-seminarky.cz/relacni-databaze/) také často používají tento typ stromu pro ukládání tabulek s indexy.

Advantages of B+ trees:

* Because B+ trees don't have data associated with interior nodes, more keys can fit on a page of memory. Therefore, it will require fewer cache misses in order to access data that is on a leaf node.
* The leaf nodes of B+ trees are linked, so doing a full scan of all objects in a tree requires just one linear pass through all the leaf nodes. A B tree, on the other hand, would require a traversal of every level in the tree. This full-tree traversal will likely involve more cache misses than the linear traversal of B+ leaves.

Advantage of B trees:

* Because B trees contain data with each key, frequently accessed nodes can lie closer to the root, and therefore can be accessed more quickly.



\*\*

The major drawback of B-Tree is the difficulty of Traversing the keys sequentially. The B+ Tree retains the rapid random access property of the B-Tree while also allowing rapid sequential access

\*\* ref: Data Structures Using C// Author: Aaro M Tenenbaum

<http://books.google.co.in/books?id=X0Cd1Pr2W0gC&pg=PA456&lpg=PA456&dq=drawback+of+B-Tree+is+the+difficulty+of+Traversing+the+keys+sequentially&source=bl&ots=pGcPQSEJMS&sig=F9MY7zEXYAMVKl_Sg4W-0LTRor8&hl=en&sa=X&ei=nD5AUbeeH4zwrQe12oCYAQ&ved=0CDsQ6AEwAg#v=onepage&q=drawback%20of%20B-Tree%20is%20the%20difficulty%20of%20Traversing%20the%20keys%20sequentially&f=false>

**ALGORITMY 179-188 DVORSKÝ**