

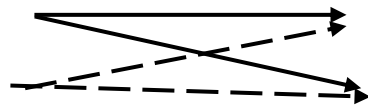
# Vyhľadávanie

Jednorozmerné

Bodové

Viacrozmerné

Intervalové



Typická štruktúra záznamu tabuľky:  $(K, S_1, S_2, \dots, S_n)$

- $K$ : *primárny* kľúč (s jedinečnými hodnotami)
- $S_i$ : *sekundárne* kľúče (ich hodnoty sa môžu opakovať)

## A. Jednorozmerné vyhľadávanie

### 1. bodové

- o podľa *primárneho* kľúča:  
nájsť (jeden) záznam pre ktorý platí:  $K = k$ , ( $k$  - hľadaný kľúč, argument)
- o podľa *sekundárneho* kľúča:  
nájsť všetky záznamy pre ktoré platí:  $S_i = s_i$  (*špeciálny prípad viacrozmerného bodového vyhľadávania*)

### 2. intervalové

- o podľa *primárneho* kľúča:  
nájsť všetky záznamy pre ktoré  $K \in [l, p]$
- o podľa *sekundárneho* kľúča:  
nájsť všetky záznamy pre ktoré platí:  $S_i \in [l, p]$  (*špeciálny prípad viacrozmerného intervalového vyhľadávania*), kde  $l$  je dolná hranica a  $p$  je horná hranica intervalu

## B. Viacrozmerné vyhľadávanie podľa sekundárnych kľúčov

Nech hľadáme záznam s kľučmi  $\{s_x, \dots, s_y\}$ .

### 1. bodové

- o nájsť všetky záznamy pre ktoré platí:  $S_x = s_x, \dots, S_y = s_y$

### 2. intervalové

- o nájsť všetky záznamy pre ktoré platí:  
 $l_x \leq S_x \leq p_x, \dots, l_y \leq S_y \leq p_y$ , kde  $l_x$  a  $p_x$  sú hranice pre sekundárny kľúč  $S_x$

Môže sa zdať, že chýba viacrozmerné vyhľadávanie podľa primárnych (jedinečných) kľúčov. Tento prípad, je ale len špecifickým prípadom viacrozmerného vyhľadávania podľa sekundárnych kľúčov.

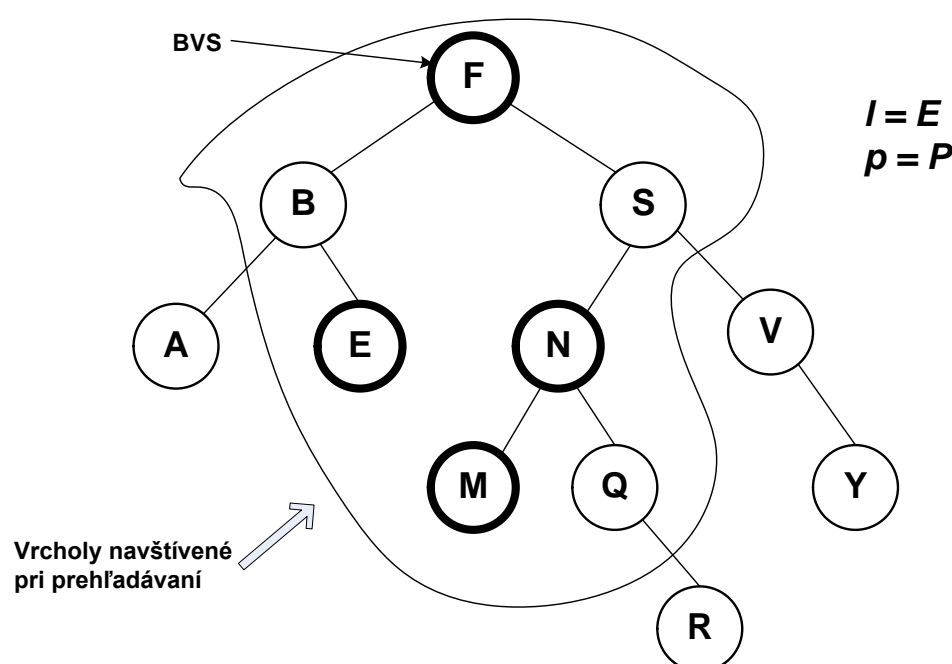
## Jednorozmerné intervalové vyhľadávanie

Doteraz Vám známe štruktúry umožňovali jednorozmerné bodové vyhľadávanie podľa primárneho kľúča (binárny vyhľadávací strom, Treap, hešovanie...). Niektoré z nich podporovali aj jednorozmerné intervalové vyhľadávanie podľa primárneho kľúča (štruktúry odvodené od binárneho vyhľadávacieho stromu).

Majme zadané  $I, p$  ( $I < p$ ) z domény kľúčov tabuľky. Úlohou je nájsť všetkých  $m$  záznamov, pre ktoré primárny kľúč  $K \in \langle I, p \rangle$ . Operáciu podporujú všetky utriedené štruktúry, teda tie, kde je možné bez nejakého dodatočného usporiadania kľúčov získať výpis prvkov usporiadaný podľa hodnoty kľúča. Na tieto štruktúry je možné aplikovať prehliadku inorder.

Najskôr sa pokúsime vyhľadať minimum z intervalu, teda  $I$ . Od tohto miesta v strome spustíme inorder prehliadku, ktorá bude ukončená okamžite po dosiahnutí vrcholu, ktorého kľuč bude väčší ako  $p$ .

Príklad algoritmu, ktorý je modifikáciou prehliadky inorder.



poradie: E F M N

**Poznámka:** Ak sa kľúč v stromovej štruktúre definuje pomocou porovnávacej funkcie (napr. interface `Comparable`, interface `Comparable`) môžeme v jej tele porovnávať aj niekoľko atribútov, teda sekundárnych kľúčov. Tu však musí platiť, že každá kombinácia kľúčových atribútov je jedinečná. Takáto implementácia, ale neznamena, že daná štruktúra podporuje viacrozmerné vyhľadávanie. Neexistuje totiž žiadna možnosť efektívne vyhľadať všetky prvky po zadaní ľubovoľného nejedinečného atribútu. Napr. máme tabuľku zamestnancov. V porovnaní sa porovnáva najskôr rok narodenia, potom meno a následne rodné číslo. Ak by sme chceli vyhľadať všetkých zamestnancov narodených v rozmedzí rokov 1979 – 1985 postupom uvedeným vyššie by to nebol problém. Toto je možné použiť iba na prvý porovnávaný atribút. Ak by sme chceli vyhľadať všetkých zamestnancov s nejakým menom, toto je možné len prejdением všetkých položiek v strome. Viacrozmerné intervalové vyhľadávanie teda nie je podporované.

Môžeme povedať, že nejaká štruktúra podporuje dané vyhľadávanie vtedy, ak sme schopní nájsť všetky hľadané záznamy bez prejdения všetkých záznamov, teda so zložitost'ou lepšou ako  $O(n)$ .

Jednorozmerné intervalové vyhľadávanie možno aplikovať aj na implicitnú (implementovanú poľom) utriedenú tabuľku:

- nájsť  $K$ , pre ktoré  $K = l$  (alebo najmenšie, pre ktoré  $K > l$ ),  $O(\log_2 n)$ ,
- prehliadať a spracovať sekvenčne, pokiaľ  $K \leq p$ . Výsledná zložitost' je  $O(m + \log_2 n)$ .