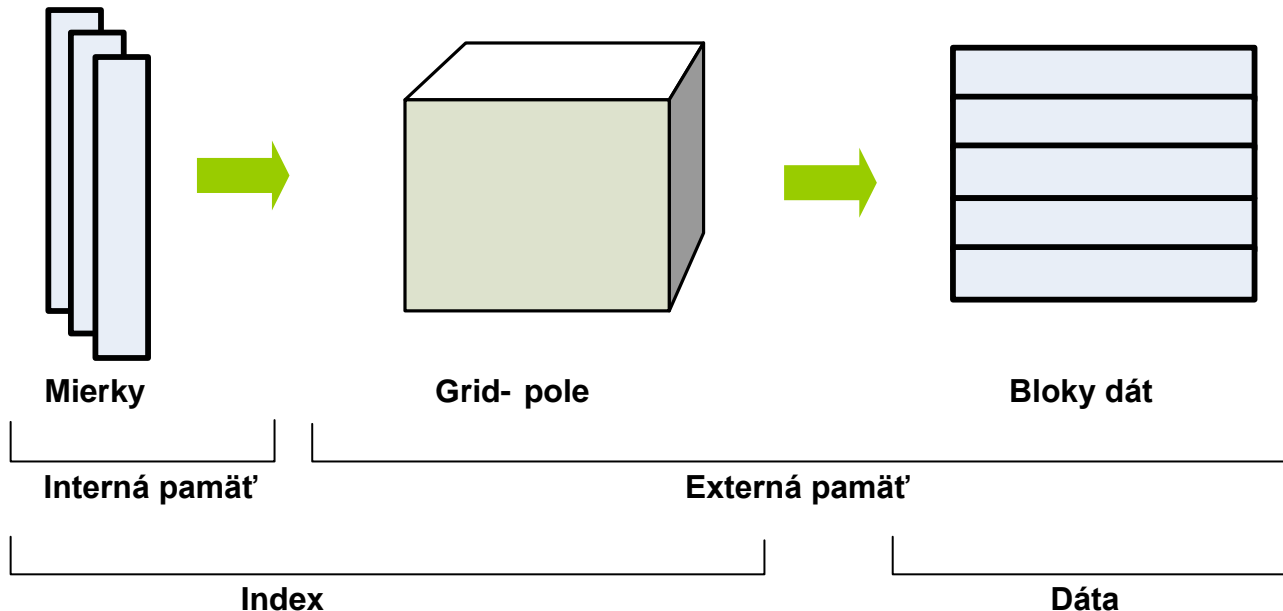


Grid - súbor

- štruktúra pre viacrozmerné intervalové vyhľadávanie - predovšetkým v externej pamäti (na disku)
- dvojúrovňová štruktúra (index, dáta)

Organizácia:



Mierka:

- pole obsahujúce rastúcu postupnosť hodnôt z domény príslušného sekundárneho kľúča $A[0] < A[1] \dots < A[m]$
- hodnoty môžu byť:
 - o ekvidištantné – rovnomerné
 - o neekvidištantné - napr. hustejšie v oblasti frekventovanejších hodnôt

0	1	2	3	4
120	140	160	180	200

Mierka „výška“ rovnomerná

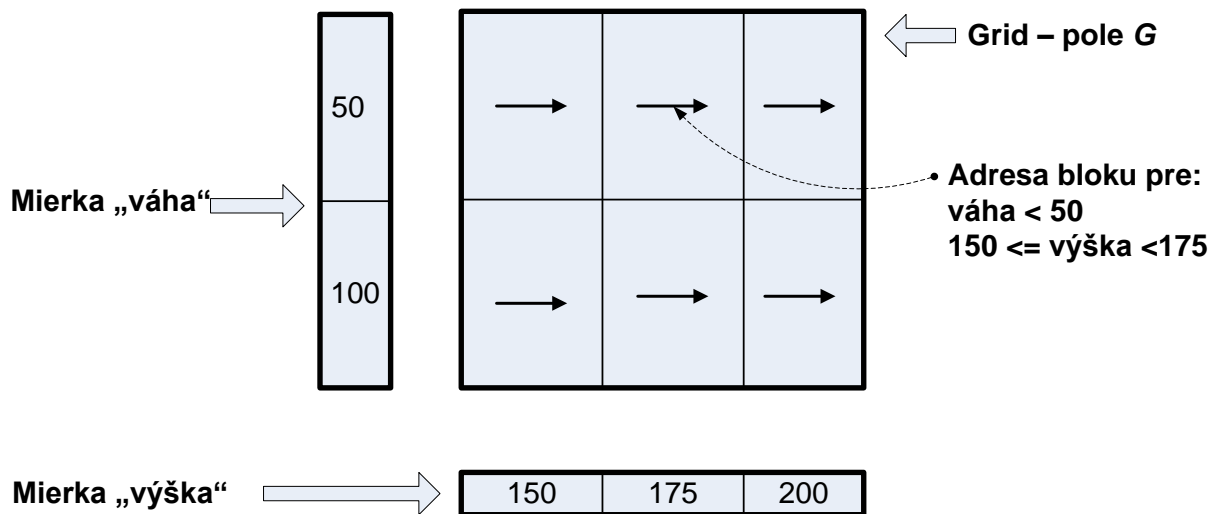
120	160	170	180	200
-----	-----	-----	-----	-----

Mierka „výška“ nerovnomerná

- pre a z príslušnej domény vieme nájsť index i taký, že: $A[i - 1] < a \leq A[i]$ alebo $i = 0$, ak $a \leq A[0]$, hovoríme, že i „pokrýva“ a
- pre a_i, a_p z príslušnej domény vieme nájsť také indexy i_i, i_p , že interval $\langle i_i, i_p \rangle$ pokrýva hodnoty z intervalu $\langle a_i, a_p \rangle$ (intervalové pokrytie)

Grid – pole:

- je to m - rozmerné pole G pre počet mierok m
- ak i pokrýva a v mierke A a j pokrýva b v mierke B , tak blok s adresou $G[i, j]$ pokrýva záznam s hodnotou sekundárnych kľúčov (a, b) , teda ak taký záznam existuje, tak je v bloku $G[i, j]$



Pamäťová reprezentácia:

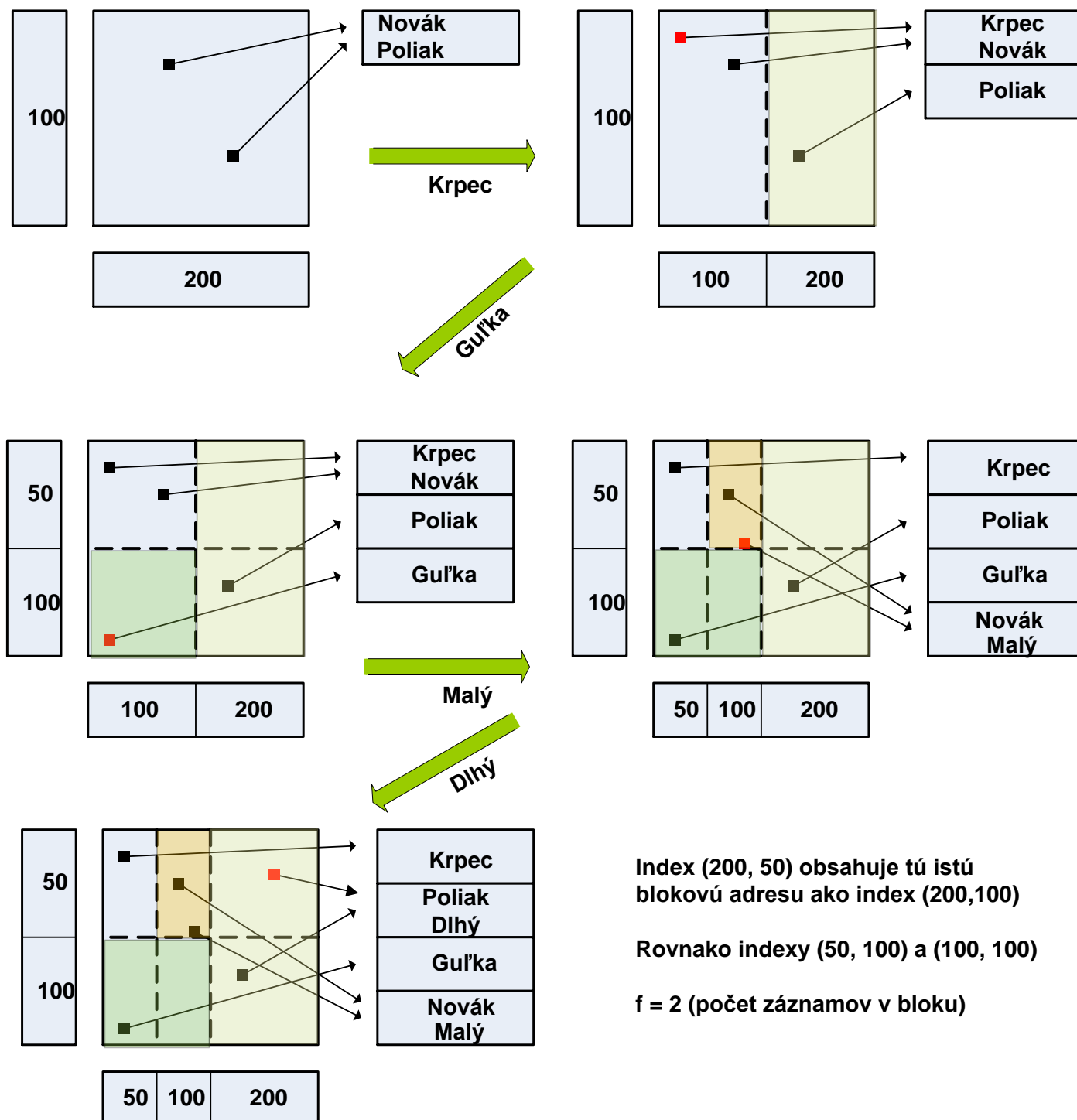
- **Mierky** - ako polia v internej pamäti
- **Grid** - pole v externej pamäti (ak je to nutné) implementované *dynamicky* (ako zoznamy adries)
- **Súbor dát** :
 - teoreticky ľubovoľná štruktúra
 - utriedené organizácie sú neefektívne, pretože grid-index umiestňuje záznamy podľa „podobnosti“
 - vhodné napr. heap (neutriedený súbor)

Find (vyhľadanie) (viacrozmerné):

- **bodové**: pokrytie v mierkach..., jedna adresa v grid-poli..., jeden blok v dátovom súbore - maximálne dva blokové prenosy
- **intervalové**:
 - nájdeme intervalové pokrytie v každej mierke
 - intervalové pokrytia v mierkach určujú viacrozmerný interval v grid-poli
 - množina prvkov tohto viacrozmerného intervalu určuje *množinu* blokov, ktoré treba prehľadať => viac blokových prenosov

Insert (vkladanie):

- bodovým vyhľadávaním nájdeme blok, do ktorého patrí
- ak blok nie je plný, vložíme bez reorganizácie
- ak je blok plný, rozdelíme ho a vložením novej (deliacej) súradnice do jednej z mierok rozdelíme aj oblasti v grid-poli



- **možné stratégie delenia:**
 - delíme striedavo zvisle a vodorovne (pri viacrozmerných cyklicky podľa jednotlivých mierok)

- delíme tak, aby v oboch podoblastiach bol rovnaký počet bodov (teda interval mierky nemusíme deliť napoly, ako je to na obrázku hore)
- reorganizácia grid poľa podstatne zvyšuje počet prenosov, preto sa používajú rôzne vylepšujúce stratégie
 - pri preplnení jedného bloku rozdeliť všetky bloky so zaplnením nad zadanú hranicu

Poznámka: Ak počet záznamov so zhodnými súradnicami (rovanskými sekundárnymi kľúčmi) prevyšuje kapacitu bloku je nutný preplňujúci blok.

Delete (mazanie):

- opačný proces ako pri vkladaní (spájanie blokov)
- keďže rekonštrukcia indexu (gridu) je najnáročnejšia, je snaha sústrediť reorganizácie (napr. urobiť všetky reorganizácie naraz, až keď hustota súboru klesne pod stanovenú hranicu)

Vlastnosti:

- ***symetričnosť*** (je jedno, či je kľúč primárny alebo sekundárny)
- ***univerzálnosť*** - zo symetričnosti plynie možnosť špeciálneho prípadu: jednorozmerná implementácia iba podľa primárneho kľúča podporujúca jednorozmerné intervalové vyhľadávanie (veľmi podobná rozšíriteľnému hešovaciemu súboru)
- ***adaptabilitnosť***: nevadia nerovnomerne rozložené dáta
- ***dobrá pomäťová lokalita***: súradnicovo blízke body viacrozmerného priestoru majú tendenciu byť v tom istom bloku

Štruktúra bola vysvetlená na 2D priestore, ale môže byť využitá ja pri viacrozmerných aplikáciách.