**Recept na COCONUT TREE indexovanie**Cieľ : Držať podobné dátové rady blízko pri sebe a zostaviť vybalancovaný index, ktorý nam urýchli hľadanie

1. Zaznamenáme jednotlivé dáta, ktoré sa uložia v jednotlivých dátových radoch

2.Dátové rady presunieme do RAM

3. Dátové rady zosumarizujeme pomocou iSAX nasledujúcim spôsobom:  
 3.1 Spravíme normalizáciu vzhľadom na normálne rozdelenie so strednou hodnotou 0 a štandardnou odchýlkou 1  
 3.2 Jednotlivé úrovne hodnôt blížiace sa k extrémom roztiahneme pre zníženie šance dosiahnutia extrémnej hodnoty, zúžime v mediánových hodnotách  
 3.3 Každý dátovy rad rozdelíme na menšie segmenty  
 3.4 Jednotlivé hodnoty v segmentoch spriemerujeme



4. Vytvoríme invSAX reprezentáciu dátových radov. Postup pre 1 rad:  
 4.1 Zoberiem Most Significant Bit z každého segmentu a zapíšem  
 4.2 Beriem ďaľšie bity a zapisujem až po Least Significant Bit  
 4.3 Príklad z obrázka vyššie: invSAX(S1)= 100100

5. Jednotlivé dátové rady zoradíme pomocou external merge sort-u podľa ich invSAX hodnoty.  
 5.Pozn.: External merge sort spája k-počet polí nachádzajúcich sa na disku, mergovanie prebieha v RAM

6. Vytvoríme index (prehladávací strom) v podobe B tree alebo B+ tree alebo UB tree  
 6.Pozn.: Vďaka tomu môžeme držať podobné dátové rady blízko a invSAX hodnnota zaznamenaná v indexe odkazuje vzároveň na umiestnenie



7. Prebehne bulk-loading (hromadné načítanie), resp. vyhradenú časť databázovej tabuľky vytiahneme spolu s dátami z RAM-ky. Tie už len rýchlo zasunieme do časti tabuľky ktorú vložíme späť