

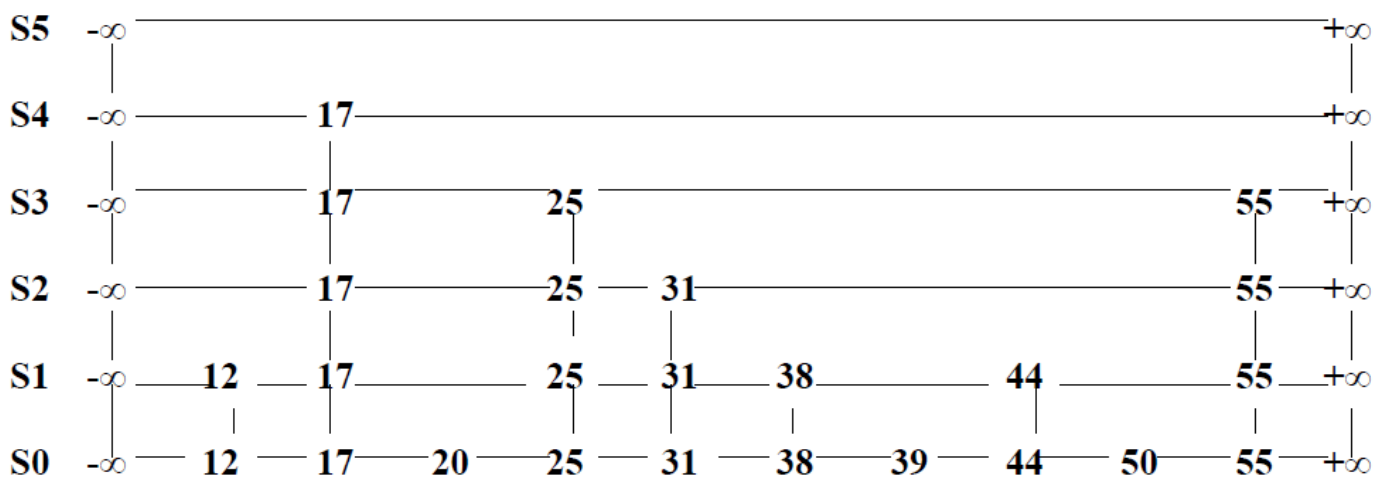
SKIP – zoznam

(Skip List)

Filozofia:

- ide o indexovanú štruktúru, čiže o štruktúru, v ktorej okrem hlavnej (dátovej) štruktúry obsahujúcej prvky (záznamy) tabuľky existuje explicitná pomocná dátová štruktúra ktorá slúži ako nástroj zefektívnenia prístupu k prvkom hlavnej štruktúry (tabuľky)
- štruktúra podporuje jednorozmerné intervalové vyhľadávanie
- hlavná štruktúra je v tomto prípade utriedená tabuľka implementovaná explicitne zoznamom.

Predstava na príklade:



Skip list:

- pozostáva z množiny zoznamov $\{S_0, S_1, \dots, S_h\}$ kde S_0 je hlavná štruktúra (tabuľka implementovaná zoznamom záznamov aj s informačnou časťou) a ostatné sú zoznamy sprístupňujúceho indexu (obsahujúce iba kľúče záznamov)
- $-\infty$ a $+\infty$ sú kľúče (zarážky) menšie resp. väčšie ako ktorýkoľvek kľúč tabuľky
- h je výška indexu
- kľúčom môže byť rovnako ako pri stromových štruktúrach akýkoľvek dátový typ
- zoznam S_h obsahuje iba zarážky

- S_i (pre $i = 1 \dots h-1$) je náhodný výber z kľúčov zoznamu S_{i-1} . Kľúč zo zoznamu S_{i-1} sa do zoznamu S_i prenesie s pravdepodobnosťou 0,5.

úroveň ... vodorovný zoznam

stípec zvislý zoznam

Predpokladajú sa 4 mechanizmy sprístupnenia prvkov, ak máme sprístupnený aktuálny prvok P : $za(P)$, $pred(P)$, $pod(P)$, $nad(p)$. Toto sprístupnenie zabezpečujú obojstranné smerníkové väzby.

Operácia nájsť kľúč K :

Polož P = prvý prvok zoznamu S_h

while $pod(P) \neq null$ do

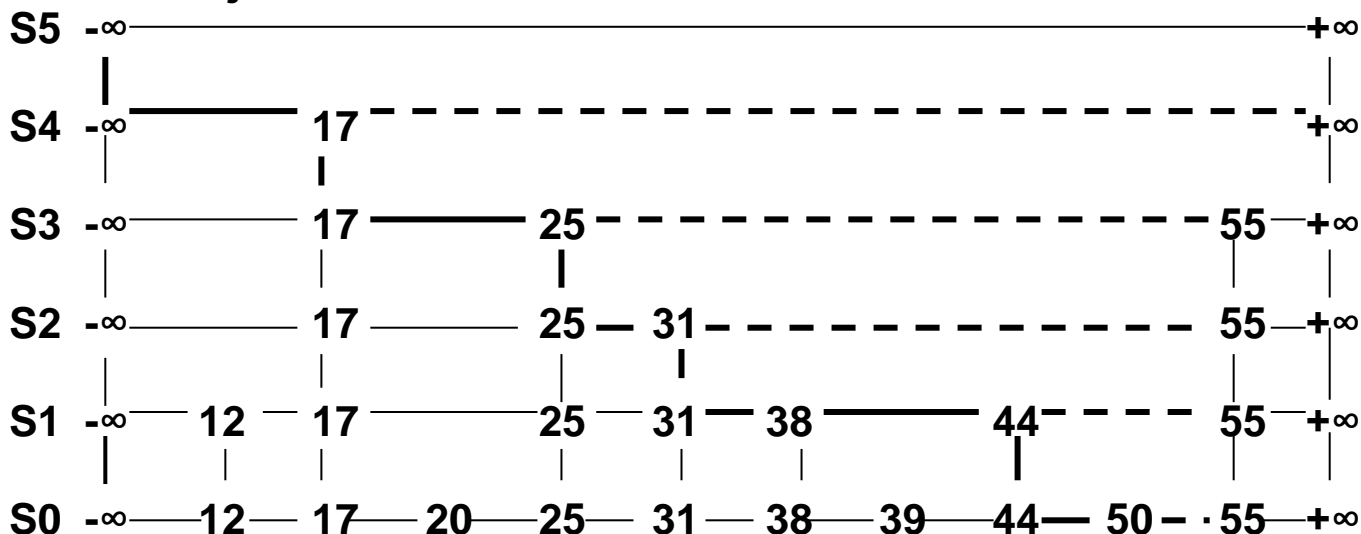
 Polož $P = pod(P)$

 while $kl'úč(za(P)) \leq K$ do polož $P = za(P)$

return (P) .

if $kl'úč(P) = K$ then úspech.

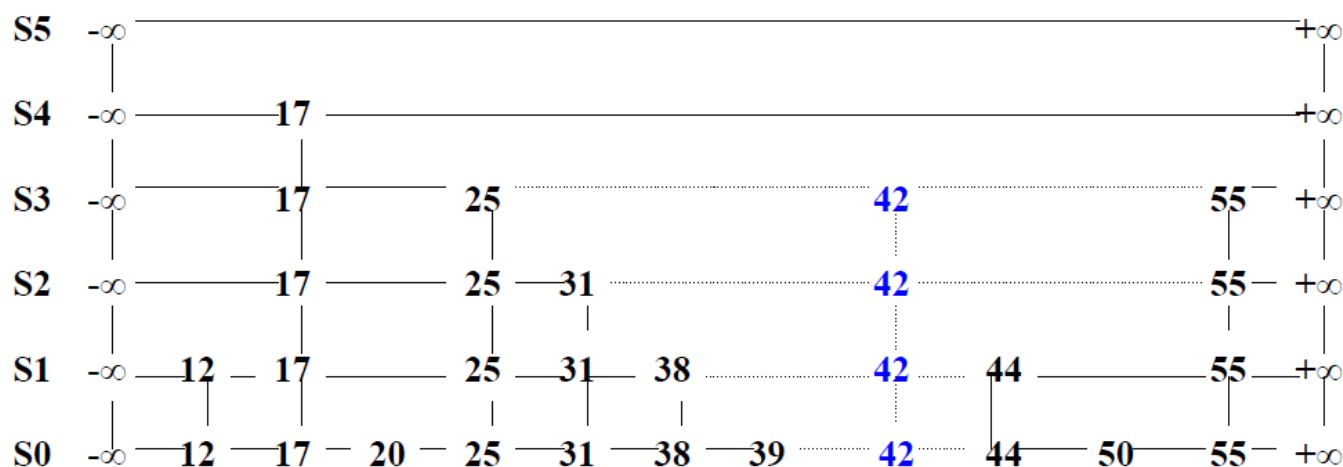
Príklad: *Nájsť kľúč 52 :*



Operácia vlož kľúč K:

- operáciou *Nájdí* (neúspešnou) nájdeme miesto vloženia : za záznam *P*
- vložíme nový záznam za záznam *P*
- nad záznam *P* vložíme stĺpec indexových záznamov s počtom záznamov r , pričom $r = 0, \dots, h-1$ podľa tohoto algoritmu:
 - a) sme na úrovni $u = 0$
 - b) s pravdepodobnosťou 0,5 vložíme indexový záznam v tomto stĺpci na úroveň $u + 1$
 - c) ak sme záznam na úroveň u vložili, tak polož $u = u + 1$, inak koniec.
 - d) ak $u \leq h$, chod' na b), inak koniec

Príklad: Vlož kľúč 42 :



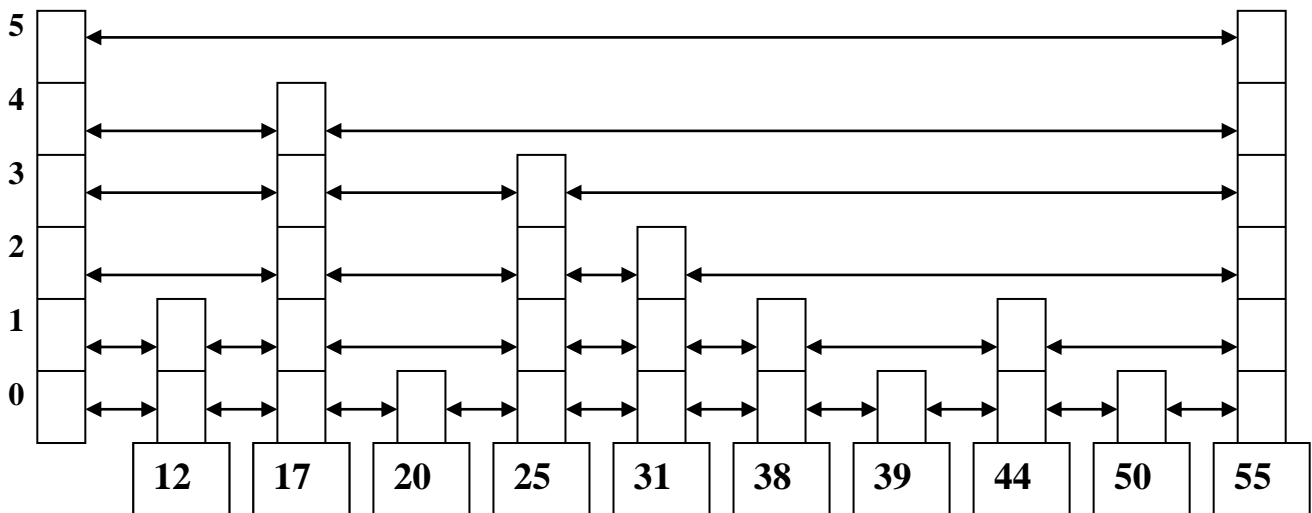
Operácia odober kľúč K:

Nájdí (K)

Odober všetky záznamy v stĺpci odpovedajúcom *K* a uprav všetky smerníky, ktoré na tento stĺpec ukazovali.

Amortizovaná (priemerná) zložitosť všetkých operácií: $O(\log_2 n)$, najhoršia $O(n)$

Implementačne výhodnejší variant Skip - zoznamu:



Hlavná zmena:

- stĺpce sú reprezentované nie zoznamom ale dynamickým poľom smerníkov (prípadne indexov) na príslušný stĺpec
- ku každému stĺpcu je evidovaný kľúč a informačná časť
- namiesto smerníkov *nad* a *pod* je pohyb v poli

Generovanie veľkosti stĺpca d :

Polož $d = 0$

while random(0, 1) < 0,5 and $d < h$ do inc(d)

return (d)

Odporúčanie: obmedziť výšku h hodnotou $\log_2 N - 1$ kde N je najväčší očakávaný počet prvkov v tabuľke.

Literatúra:

<http://www.geocities.com/SiliconValley/Network/1854/skiplist.html>

<http://ww3.java2.datastructures.net/presentations/SkipLists.pdf>