

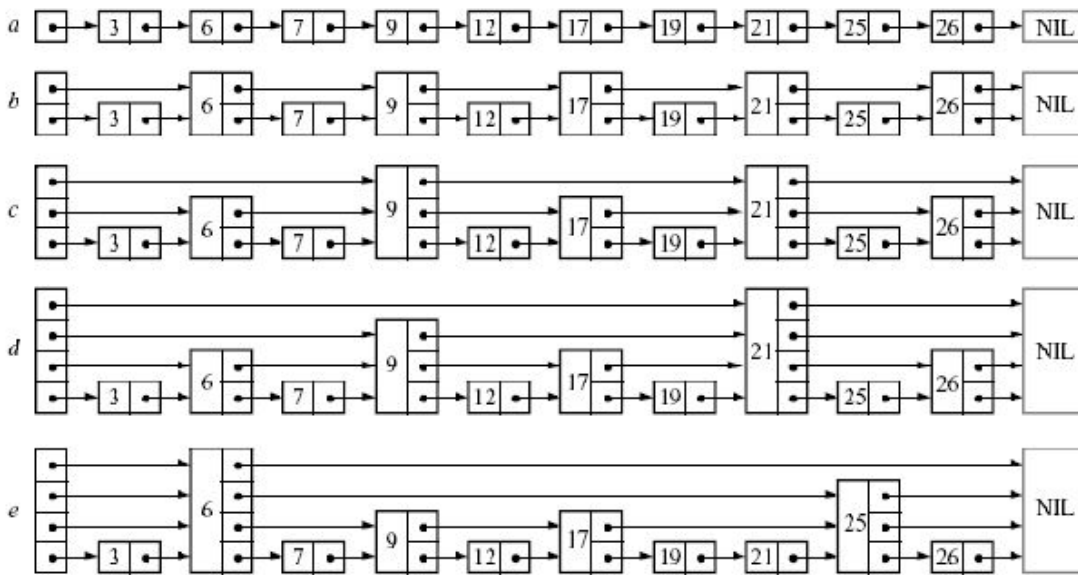
Skip-list - použití a implementace

Z FAV wiki

Skiplist je alternativa ke stromům, která je snadno implementovatelná. Operace se skiplistem jsou podobné složitosti jako u stromů ($O(n \log n)$), ale paměťová náročnost může být o něco větší.

Skiplist si lze představit následovně: seznam, ve kterém některé prvky ukazují i na vzdálenější prvky než jsou prvky bezprostředně následující. Tyto ukazatele "přeskakují" položky seznamu, odtud skiplist.

Maximální výška uzlu = Maxlevel, každý prvek může mít $1 \leq k \leq \text{Maxlevel}$ ukazatelů na další prvky



Popisky (obrázek je z přednášky):

- **a)** vidíme nejjednodušší skiplist se skokem 1, tedy lineární spojový seznam
- **b)** skiplist se skokem 2. Každý druhý prvek obsahuje kromě hodnoty a ukazatele na následníka také ukazatel na následníka následníka ($\text{next}[\text{next}[x]]$). Už touto změnou zkrátíme vyhledávání na polovinu, jelikož abychom se dostali na prvek např. 25, stačí nám pouze 5 skoků ($> 6, > 9, > 17, > 21, > 25$) na rozdíl od seznamu, kde jich je 9
- **c)** skiplist se skokem 4. na prvek 25 se dostaneme pouze 2 skoky
- **d)** skiplist se skokem 8. na prvek se dostaneme 1 skokem \Rightarrow ideální skip-list: každý 2i-tý prvek má ukazatel, který ukazuje o 2i prvků dopředu
- **e)** skiplist s náhodným skokem - ?? skoků, ale ve velkém skiplistu je rychlejší, než lineární seznam, a operace přidání a odebrání jsou jednodušší

Výhody:

- Skiplist je extrémně rychlý (vyhledávání je $\log n$)
- Velmi snadná implementace (pouze pole ukazatelů v každém prvku)

Nevýhody:

- Pomalé operace insert a delete (je nutné reorganizovat skiplist)

Řešení: skiplist s náhodnými velikostmi uzlů. Potom:

- Vložení - najdeme místo, kam prvek patří, pomatujeme si všechny předchůdce. Po nalezení vložíme prvek s náhodnou výškou $1 < k < \text{Maxlevel}$, a napojíme všechny předchůdce do výšky tohoto prvku (samozřejmě na obě strany, předchůdci ukazují na tento prvek, a ten ukazuje na prvky, na které předchůdci ukazovali původně)
- Mazání - opačný proces, najdeme prvek a pomatujeme si předchůdce. Pro všechny předchůdce do výšky hledaného prvku

upravíme ukazatele (předchůdci budou ukazovat na prvky, na které ukazoval hledaný prvek). Hledaný prvek smažeme.

Tento skip-list není zcela $\log n$, a nelze (díky náhodnosti) jeho složitost určit. S vysokou pravděpodobností však všechny operace budou rychlejší než $O(n)$ (samozřejmě můžeme mít smůlu a všechny prvky budou náhodně výšky 1).

Citováno z „http://www.512.cz/index.php?title=Skip-list_-_pou%C5%BEit%C3%AD_a_implementation“

Kategorie: Fav-kiv-bzinf

- Stránka byla naposledy editována 20. 2. 2014 v 06:44.
- Stránka byla zobrazena 1 496krát.