Napredni algoritmi i strukture podataka

SimHash, SkipList

Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka

SimHash - algoritam

- Set podataka (npr. tekst) podelimo na delove i uklonite zaustavne reči (ako ih ima)
- Dodelimo težine dobijenim vrednostima (npr. broj ponavljanja reči)
- Izračunamo **b**-bitni *hash* za svaki element iz dobijenog skupa, propuštajući element kroz *hash* funkciju
- Za svaku dobijenu vrednost uradimo konverziju $\mathbf{0} \rightarrow \mathbf{-1}$
- Formiramo tabelu, tako što vrednosti stavimo jedne ispod drugih
- Sumiramo kolone, množeći težine sa vrednošću
- Ponovo izvršimo konverziju, ali sada za svaku vrednost u dobijenom rezultatu:
  - if $el > 0$, $el \leftarrow \mathbf{1}$
  - if $el < 0$, $el \leftarrow \mathbf{0}$
- Dobijamo **b**-bit fingerprint za ceo ulazni set
- Uradimo XOR operaciju sa drugim setom podataka i dobijamo Hemingovu udaljenost

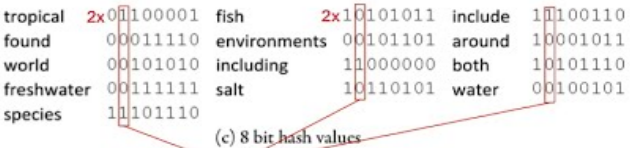
SimHash - primer

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

(a) Original text

tropical 2 fish 2 include 1 found 1 environments 1 around 1 world 1
including 1 both 1 freshwater 1 salt 1 water 1 species 1

(b) Words with weights



(c) 8 bit hash values

1 -5 9 -9 3 1 3 3

(d) Vector V formed by summing weights

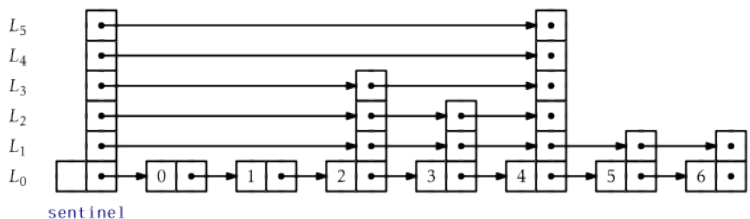
1 0 1 0 1 1 1 1

(e) 8-bit fingerprint formed from V

(Victor Lavrenko, SimHash)

Skip list

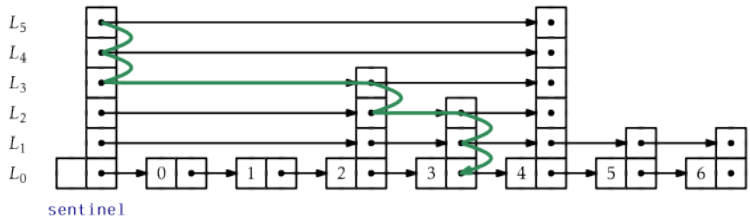
- Možete zamisliti ovu strukturu kao sistem metroa
- Postoje vozovi koji staje na svakoj stanici
- Ali, postoji i ekspresni voz koji staje na manje stanica
- Ovo čini ekspresni voz atraktivnom opcijom ako znate gde staje



Skip list - pretraga

Pretraga elementa **k** se vrši po sledećem algoritmu

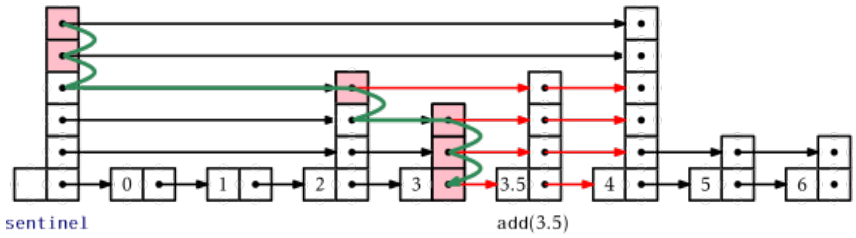
- Ako je k = key, kraj
- Ako je k < next key, prelazimo na nivo ispod
- Ako je k >= next key, idemo desno



Skip list - dodavanje

- Lociramo gde bi element trebalo da se doda — **neuspešna pretraga**
- Povežemo pokazivač prethodnog elementa sa novokreiranim elementom
- Pokazivač novokreiranog elementa pokazuje na naredni element
- Ove operacije su identične kao i kod linked liste
- **ALI**, treba i da odredimo koliko nivoa naš element ima i za svaki nivo treba ispravno da povežemo pokazivače
  - Dodajemo element u nivo **0**
  - **while** $\mathrm{FLIP}() == \mathrm{'GLAVA'}$
    - Dodajemo novi nivo
    - Povećavamo nivo elementa

Skip list - dodavanje

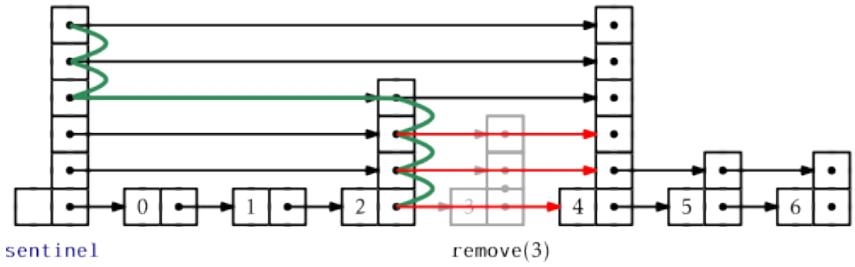


Skip list - brisanje

Brisanje elementa **k** se vrši po sledećim koracima:

- Lociramo koji element trebalo da se obriše — **neuspešna pretraga**
- Kada je element lociran, prevezujemo pokazivače da bi se element uklonio iz liste, baš kao što radimo u linked listi.
- Brisanje počinjemo od najnižeg nivoa i vršimo prevezivanje pokazivača sve dok ne stignemo do elementa
- Nakon brisanja elementa može postojati nivo bez elemenata, tako da ćemo i ove nivoe ukloniti, smanjivši nivo Skip liste.

Skip list - brisanje



Zadaci

- Implementirati SimHash algoritam
- Implementirati SkipList strukturu