Hešovacie funkcie

- Zobrazenie z veľkej množiny prvkov do malej množiny kľúčov.
- Dobré sa správajú „náhodne“.
  (To vieme aj matematicky zaručiť, ak máme skill.)
- Rovnomerne rozhádžeme uložené prvky do vedierok.
- Treba riešiť kolízie.
- Treba dosť vedierok (aspoň lineárne veľa).

Hešovacie funkcie

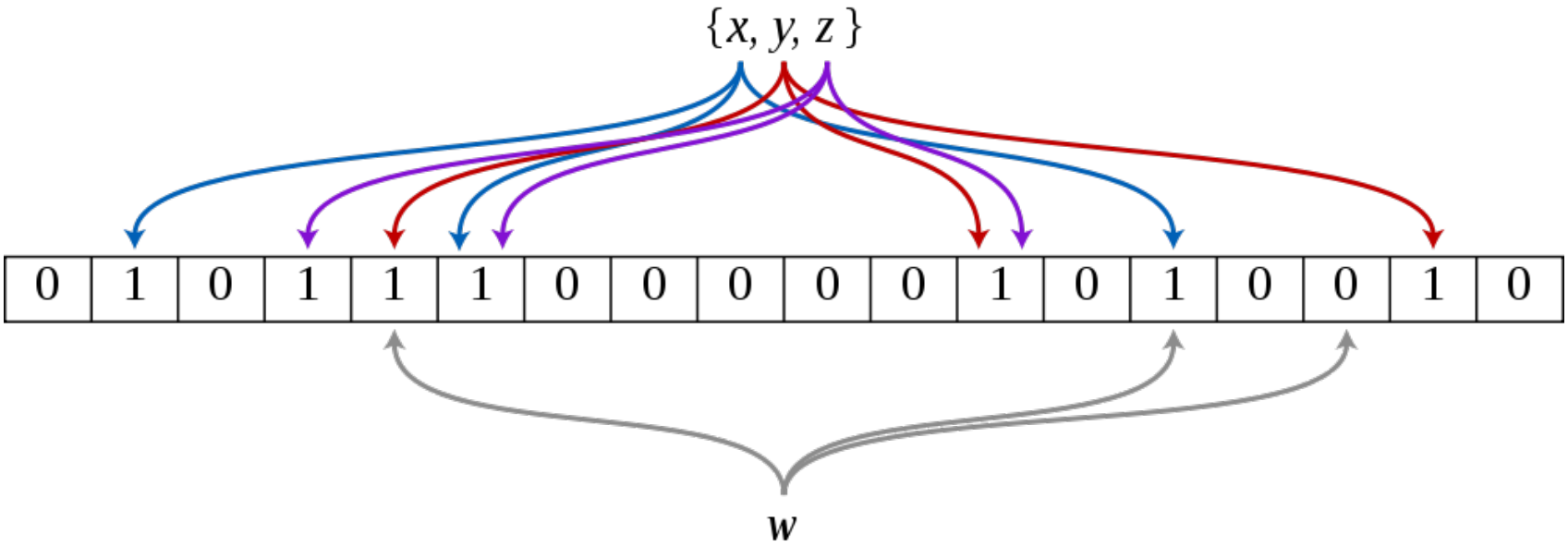
- Zobrazenie z veľkej množiny prvkov do malej množiny kľúčov.
- Dobré sa správajú „náhodne“.
  (To vieme aj matematicky zaručiť, ak máme skill.)
- Rovnomerne rozhádžeme uložené prvky do vedierok.
- Treba riešiť kolízie.
- Treba dosť vedierok (aspoň lineárne veľa).

Hešovacie funkcie

- Zobrazenie z veľkej množiny prvkov do malej množiny kľúčov.
- Dobré sa správajú „náhodne“.
  (To vieme aj matematicky zaručiť, ak máme skill.)
- Rovnomerne rozhádžeme uložené prvky do vedierok.
- Treba riešiť kolízie.
- Treba dosť vedierok (aspoň lineárne veľa).

Bloomov filter

- „Hešovacia tabuľka“, no nepamätáme si ani len samotné prvky.
- Napriek tomu vieme vkladať aj skoro s istotou testovať.
- Použitý napr. v Google BigTable.



Bloomov filter

Cool fičúrie navyše

- V konečnej pamäti viem uložiť množinu obsahujúcu všetko.
- Funguje bitwise and/or na prienik/zjednotenie.

Pamäťové nároky

- False positive pp (k hashov, m bitov, n prvkov): približne $(1-(1-1/m)^{kn})^k \approx (1-e^{-kn/m})^k$
- Konkrétne príklady:
  $k=7$, $m/n=10$: zhruba $1/100$ false positive
  $k=9$, $m/n=14$: zhruba $1/1000$ false positive

Bloomov filter

Cool fičúrie navyše

- V konečnej pamäti viem uložiť množinu obsahujúcu všetko.
- Funguje bitwise and/or na prienik/zjednotenie.

Pamäťové nároky

- False positive pp (k hashov, m bitov, n prvkov): približne $(1 - (1 - 1/m)^{kn})^k \approx (1 - e^{-kn/m})^k$
- Konkrétne príklady:
  $k = 7$, $m/n = 10$: zhruba $1/100$ false positive
  $k = 9$, $m/n = 14$: zhruba $1/1000$ false positive