#5 Hešujem, hešuješ, hešujeme

©kuko 16.10.2019

1 Hešovanie

- hešovacia tabuľka veľkosti n, m prvkov, hešovacia fn. $h : U \to \mathbb{Z}_n$
- ako budeme riešiť kolízie?
  - zreťazením? – zlé kvôli cache missom
  - ďalšou heš-tabuľkou (viď perfektné hešovanie)?
  - lineárnym sondovaním? – rýchle v praxi
- ako vybrať hešovaciu fn.?
  - v praxi batéria testov, pozri `https://github.com/rurban/smhasher`
  - SipHash (Python, ruby, rust, haskell, ...), CityHash (používaný v Googli – Abseil), xxHash, Murmur, FNV

2 Riešenie kolízií zreťazením

- hodnoty, ktoré sa zahešujú na jedno políčko uložíme do spájaného zoznamu – takto je implementovaný `unordered_set`/`unordered_map` v STL
- model guličiek a krabíc: ak predpokladáme, že hádžeme m guličiek do n krabíc úplne náhodne a nezávisle, dá sa dokázať:
  - na jedno políčko sa zahešuje v priemere m/n prvkov
  - ale na najvyťaženejšom políčku je s vysokou pp. $\Theta(\log n / \log\log n)$ prvkov pre $m = n$
  - zhruba $ne^{-m/n}$ políčok je prázdnych (viac ako tretina pre $m = n$)
  - kolíziu dostaneme už pri $m \sim \sqrt{n}$ (narodeninový paradox: $\prod_{i=1}^{m}(1 - i/n) \leq \prod e^{-i/n} = 1/e^{m(m+1)/2n}$)
- protivník však ľahko nájde vstup, na ktorom všetky prvky padnú na to isté políčko
- namiesto spoliehania sa na náhodný vstup môžeme vybrať náhodnú *hešovaciu funkciu*; takto premeníme priemerný prípad na očakávaný

- rodina hešovacích funkcií $\mathcal{H}$ je k-univerzálna (k-nezávislá), ak $\forall x_1, \ldots, x_k$ všetky rôzne a $\forall y_1, \ldots, y_k$, ak zvolíme náhodnú hešovaciu funkciu $h \in_R \mathcal{H}$, tak

$$\Pr_{h \in \mathcal{H}}[h(x_1) = y_1, h(x_2) = y_2, \ldots, h(x_k) = y_k] = O(1/n^k).$$

- inými slovami $\forall x, y : \Pr_h[h(x) = y] = O(1/n)$ a pre rôzne $x_1, \ldots, x_k$ sú náhodné premenné $h(x_1), \ldots, h(x_k)$ skoro nezávislé
- napríklad: $x \mapsto (ax \bmod p) \bmod n$, $x \mapsto (ax) >> (\lg u - \lg n)$ (ak u, n sú mocniny 2); sú (1-)univerzálne
- $x \mapsto ((ax + b) \bmod p) \bmod n$ sú 2-univerzálne, všeobecne $x \mapsto ((a_k x^k + \cdots + a_1 x + a_0) \bmod p) \bmod n$ sú k-univerzálne
- jednoduché tabulačné hešovanie: vygenerujeme si tabuľky $T_1, \ldots, T_c$ veľkosti $u^{1/c}$ s úplne náhodnou hešovacou fn.;
- na $x \in U$ sa pozeráme ako na vektor $x = x_1 \cdots x_c$ a $h(x) = T_1(x_1) \oplus \cdots \oplus T_c(x_c)$
- pre jednoduché tabulačné hešovanie sa dá dokázať veľa výsledkov, ktoré platia pre $O(\log n)$-nezávislé rodiny
- na heš-tabuľky, kde kolízie riešime zreťazením stačí univerzálna rodina $\mathcal{H}$: E[dĺžka reťaze] $= \sum_i \Pr[h(x_i) = t] = m \times O(1/n) = O(1)$ pre $m = \Theta(n)$

3 Perfektné hešovanie

- problém: statický slovník, t.j. záznamy sa nemenia (nevkladajú ani nevymazávajú)
- [Fredman, Komlós, Szemerédi '84]: očak. $O(n)$ zostrojenie, vyhľadávanie v $O(1)$ det.
- namiesto reťazenia použijeme na druhú úroveň hešovaciu tabuľku kvadratickej(!) veľkosti
- očak. #kolízií je $m^2 \cdot O(1/n) = O(1) \leq 1/2$ pre dosť veľké $n = \Theta(m^2)$
- z Markovovej nerovnosti nemáme žiadnu kolíziu s pp. aspoň 1/2
- $\mathrm{E}[\sum_t C_t^2] = \sum_t \mathrm{E}[C_t^2] =$ očakávaný počet dvojíc, ktoré kolidujú $= \sum_{i,j} \Pr[h(x_i) = h(x_j)] = O(m^2/n) = O(n)$ pre $m = \Theta(n)$
- dá sa rozšíriť na dynamický slovník s vkladaním a vymazávaním v $O(1)$ očakávane amort. (dá sa dokonca zariadiť $O(1)$ s vysokou pp. amortizovane), ale algoritmus je nepraktický

4 Kukučie hešovanie

- máme 2 tabuľky A, B dĺžky $2m$ a 2 hešovacie funkcie f, g
- hľadanie: prvok x je vždy v $A[f(x)]$ alebo $B[g(x)]$
- vkladanie:
  - ak je $A[f(x)]$ alebo $B[g(x)]$ voľné, dáme ho tam
  - ak nie, prvok y v A vyhodíme a pokúsime sa vložiť y do $B[g(y)]$
  - ak v $B[g(y)]$ je prvok z, vyhodíme ho a skúsime ho vložiť do $A[f(z)]$, atď.
  - ak sa zacyklíme, fail: vyberieme novú hešovaciu fn. a celú tabuľku prebudujeme
- úplne náhodné alebo $O(\log n)$-nezávislé – očak. amort. zložitosť $O(1)$, pp. failu $O(1/n)$
- (btw: 6-nezávislosť nestačí; jednoduché tabulačné hešovanie: pp. failu $O(1/\sqrt[3]{n})$)
- intuitívne, ak sú A aj B plné max do polovice, máme vždy zhruba 1/2 pravdepodobnosť, že nájdeme prázdne políčko a $\Pr[\text{insert prejde cestu dĺžky } k] \leq 1/2^k$

5 Lineárne sondovanie

- máme tabuľku veľkosti $n \geq (1+\varepsilon)m$; ak je pozícia $h(x)$ obsadená, skúšame $h(x)+1, h(x)+2, \ldots$
- zlá povesť kvôli lineárnemu clustrovaniu
- v praxi najlepšie (kvôli cachovaniu) – kvadratické sondovanie je blbosť
- [Pagh, Pagh, Ružić '07] stačí 5-nezávislosť
- [Pătrasçu, Thorup '10] 5-nezávislosť treba (existuje rodina 3-, 4-nezávislých heš. fn., taká, že očakávaná zložitosť je $\Theta(\log n)$, pre 2-nezávislé dokonca $\tilde{\Theta}(\sqrt{n})$)
- majme $n = 3m$ (v skutočnosti stačí predpokladať $n = (1+\varepsilon)m$; dostaneme zložitosť $O(1/\varepsilon^2)$)
- predstavme si nad hešovacou tabuľkou kompletný binárny strom
- vrchol vo výške h má pod sebou 2^h políčok a v priemere sa doň zahešuje $\mu = \frac{1}{3}2^h$ kľúčov
- vrchol budeme volať *nebezpečný*, ak sa do intervalu pod ním zahešuje aspoň $\frac{2}{3}2^h$ kľúčov ($\geq 2\mu$)
- pozor, miesto $h(x)$, kam sa x zahešuje a miesto, kde x nakoniec skončí (kvôli tomu, že niektoré políčka sú už obsadené) sú dve rôzne veci; nás zaujíma to prvé – $h(x)$
- keďže $n = 3m$, pp., že vrchol je nebezpečný je $\Pr[\#\text{kľúčov} \geq 2\mu] = (e/4)^\mu$ (Černofova nerov.)
- pozrime sa na jeden beh (cluster) zaplnených políčok dĺžky $[2^\ell, 2^{\ell+1})$
- takýto interval je celý pokrytý 4–9 vrcholmi vo výške $\ell - 2$; (8 vrcholov pokryje $2^{\ell+1}$ políčok, ale interval môže byť posunutý)
- tvrdíme, že aspoň jeden z nich musí byť nebezpečný
- sporom: predpokladajme, že všetky vrcholy sú bezpečné (hešuje do nich $< \frac{2}{3}2^{\ell-2}$ kľúčov)
- pozrime sa na prvé tri vrcholy: aj keby z prvého vrcholu všetky kľúče „pretiekli“ doprava, v ďalších dvoch vrcholoch je dosť miesta ($> \frac{2}{3}2^{\ell-2}$) aby ich absorbovali
- t.j. ak by prvé tri vrcholy boli bezpečné, v behu by bola diera – spor
- $\Pr[x \in \text{beh dĺžky } [2^\ell, 2^{\ell+1})] \leq 3\Pr[\text{vrchol vo výške } \ell-2 \text{ je nebezpečný}] \leq 3(e/4)^{2^{\ell-2}}$
- $\mathrm{E}[\text{zložitosť operácie}] = \sum_\ell O(2^\ell) \cdot \Pr[x \in \text{beh dĺžky } [2^\ell, 2^{\ell+1})]) = \Theta(1)$