Data Structures and Algorithms

István András Seres @Eötvös Loránd University

2019 April 30, Tuesday 12:30-14:00, D-0.221

1 Separate chaining, linear and quadratic probing

Consider inserting the following keys into a hash table of length $m = 10$:

$$2, 4, 1, 20, 15, 8, 31, 14, 3, 0, 11, 28$$

The auxiliary hash function is given by $(k \mod m)$. Draw the resulting hash table if we use separate chaining, linear and quadratic probing with parameters $c_1 = 2$ and $c_2 = 3$.

2 Efficiently deciding subsets

Consider two sets of integers, $S = (s_1, s_2, ..., s_m)$ and $T = (t_1, t_2, ..., t_n)$, $m \leq n$.

- Device an algorithm that uses a hash table of size m to test whether S is a subset of T.
- What is the average running time of your algorithm?

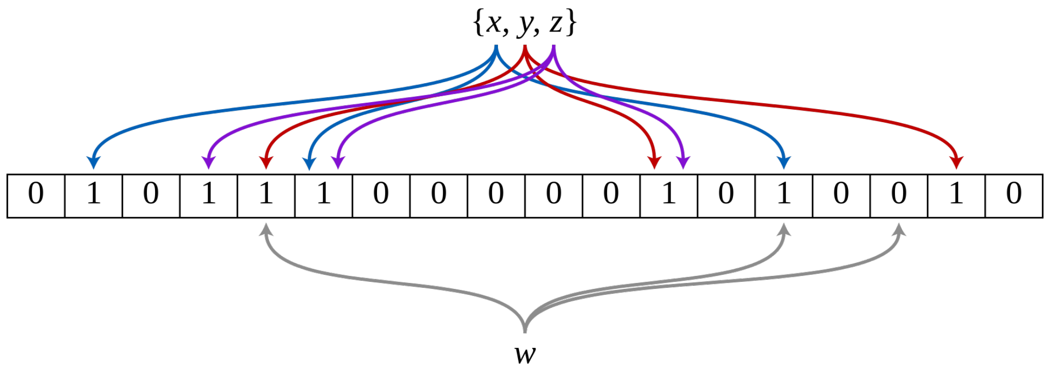
3 Delete in hash tables

Suggest how to implement $Delete(k)$ function for a hash table, when using open addressing.

4 Bloom Filters

An empty Bloom filter is a bit array of m bits, all set to 0. There must also be k different hash functions defined, each of which maps or hashes some set element to one of the m array positions, generating a uniform random distribution. Typically, k is a constant, much smaller than m, which is proportional to the number of elements to be added; the precise choice of k and the constant of proportionality of m are determined by the intended false positive rate of the filter.

To add an element, feed it to each of the k hash functions to get k array positions. Set the bits at all these positions to 1.



To query for an element (test whether it is in the set), feed it to each of the k hash functions to get k array positions. If any of the bits at these positions is 0, the element is definitely not in the set – if it were, then all the bits would have been set to 1 when it was inserted. If all are 1, then either the element is in the set, or the bits have by chance been set to 1 during the insertion of other elements, resulting in a false positive.

Show that the probability of a false positive is:

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \simeq \left(1-e^{-\frac{kn}{m}}\right)^{k}$$

5 Piros-fekete fák egyesítése

Legyenek F_1 és F_2 piros-fekete fák. Tegyük fel, hogy az F_2-ben szereplő kulcsok mind nagyobbak az F_1-ben szereplő kulcsoknál. Mutassuk meg, hogy a két piros-fekete fa $O(\log n)$ költséggel összefűzhető egyetlen piros-fekete fává, ahol n a két fában szereplő kulcsok együttes száma!

6 Még egyszer a k-adik elem kiválasztásáról

Valós számok egy véges halmazát szeretnénk tárolni olyan adatszerkezettel, amely a szokásos keres, beszúr, töröl műveleteken kívül támogatja a halmaz k-adik elemének kiválasztását. Valósítsuk meg minél hatékonyabban az adatszerkezetet!