Day 26 – Randomize Algorithms (cont.)

CSE21 Fall 2018

December 5, 2018

https://sites.google.com/ucsd.edu/cse21fa18/

Element Distinctness

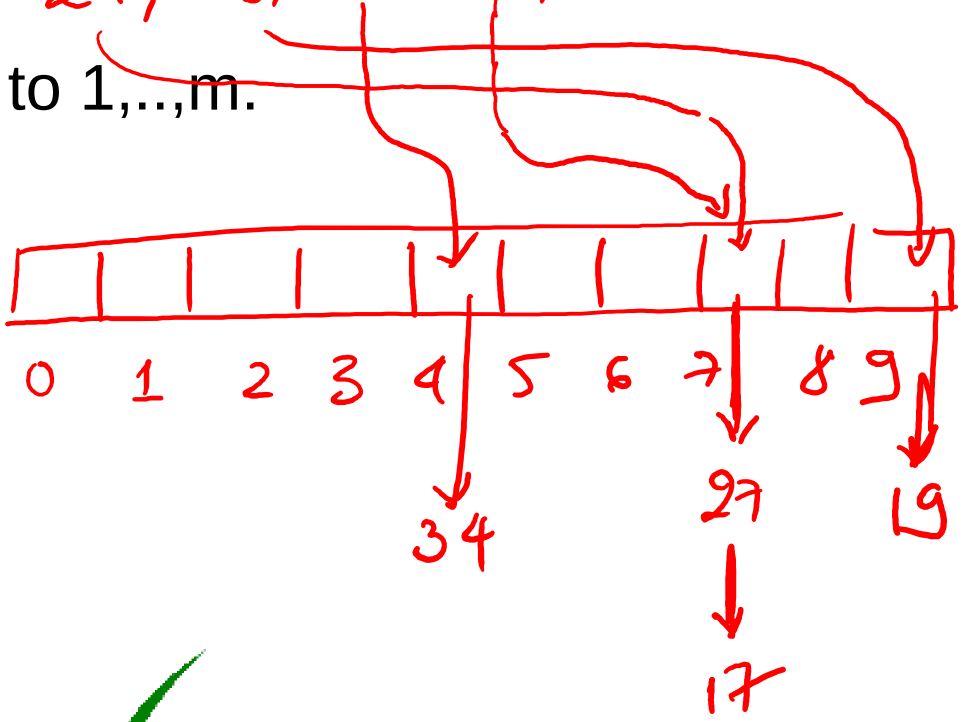
Given list of positive integers A = $a_1, a_2, \ldots, a_n$, and m memory locations available

h = mod 10

List: 27, 19, 34, 17, ...

ChainHashDistinctness(A, m)

1. Initialize array M[1,..,m] to null lists.
2. Pick a hash function h from all positive integers to 1,..,m.
3. For i = 1 to n,
4. For each element j in M[$h(a_i)$],
5. If $a_j = a_i$ then return "Found repeat"
6. Append a_i to the tail of the list M [$h(a_i)$]
7. Return "Distinct elements"



***Correctness:* Goal is**

If there is a repetition, algorithm finds it

If there is no repetition, algorithm reports "Distinct elements"

Element Distinctness: MEMORY

Given list of positive integers A = a_1, a_2, …, a_n , and m memory locations available

ChainHashDistinctness(A, m)

1. Initialize array M[1,..,m] to null lists. *
2. Pick a hash function h from all positive integers to 1,..,m.
3. For i = 1 to n,
4. For each element j in M[h(a_i)],
5. If a_j = a_i then return "Found repeat"
6. Append a_i to the tail of the list M [h(a_i)]
7. Return "Distinct elements"

What's the memory use of this algorithm?

Size of memory locations: $O(m)$. Total size of all the linked lists: $O(n)$.

Total memory: $O(m+n)$.

Element Distinctness: WHEN

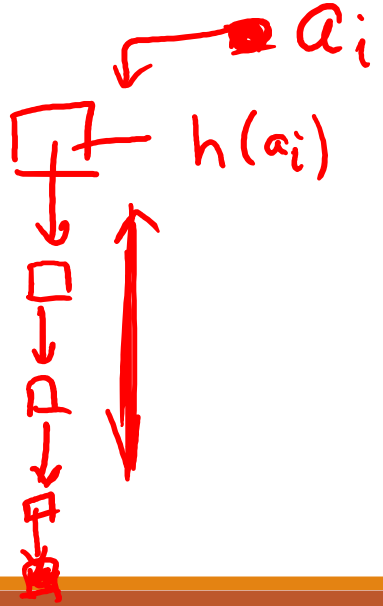
ChainHashDistinctness(A, m)

1. Initialize array M[1,..,m] to null lists.
2. Pick a hash function h from all positive integers to 1,..,m.
3. For i = 1 to n,
4. For each element j in M[$h(a_i)$],
5. If $a_j = a_i$ then return "Found repeat"
6. Append a_i to the tail of the list M [$h(a_i)$]
7. Return "Distinct elements"

$\Theta(1)$ (lines 1–3)

Worst case is when we don't find a_i:
O(1 + size of list M[$h(a_i)$])
= O(1 + # j<i with $h(a_j)=h(a_i)$)

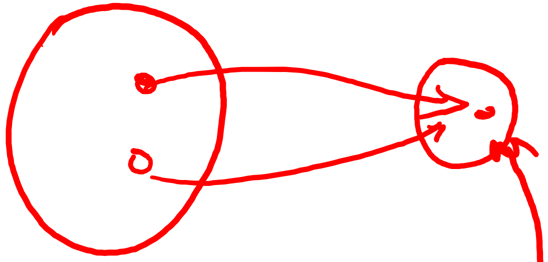
$\Theta(1)$ (line 7)



Total time: O(n + $\sum_{i=1}^{n}$ # collisions between pairs a_i and a_j, where j<i)

= **O(n + total # collisions)**

Element Distinctness: WHEN



Collisions depend on choice of **hash function**

h: { desired memory locations } → { actual memory locations }

Ideal hash function model: each output in {1,2,…,m} is equally likely.

So **h** is a function that chooses a random number in {1,2,…,m} for each input a_i.

Element Distinctness: WHEN

Total time: O(n + $\sum_{i=1}^{n}$ # collisions between pairs a_i and a_j, where j<i)

= **O(n + total # collisions)**

What's the expected total number of collisions? $\Rightarrow$ $E(X)$

X

For each pair (i,j) with j<i, define: **$X_{i,j}$ = 1 if $h(a_i)=h(a_j)$ and $X_{i,j}$=0 otherwise.**

Total # of collisions = $\sum_{(i,j):j<i} X_{i,j}$

$E(X_{ij}) = (1) \cdot P(X_{ij}=1) + \cancel{(0) P(X_{ij}=0)}$

$= \frac{1}{m}$

So by linearity of expectation: E(total # of collisions) = $\sum_{(i,j):j<i} E(X_{i,j})$

$\binom{n}{2}$ pairs

Element Distinctness: WHEN

Total time: O(n + $\sum_{i=1}^{n}$ # collisions between pairs a_i and a_j, where j<i)

= **O(n + total # collisions)**

What's the expected total number of collisions?

For each pair (i,j) with j<i, define: **$X_{i,j}$ = 1 if $h(a_i)=h(a_j)$ and $X_{i,j}$=0 otherwise.**

So by linearity of expectation:

E(total # of collisions) = $\sum_{(i,j):j<i} E(X_{i,j}) = \binom{n}{2}\frac{1}{m} = O(n^2/m)$

Element Distinctness: WHEN

Total time: $O(n + \sum_{i=1}^{n}$ # collisions between pairs a_i and a_j, where $j<i$)

= $O(n +$ total # collisions)

Total expected time: $O(n + n^2/m)$

In ideal hash model, as long as $m>n$ the total expected time is **$O(n)$.**

Note: This is much better than our original approach using sorting.

$\hookrightarrow O(n \log n)$

Birthday paradox

Given a group of n people. Assume that each person is equally likely to have any birthday. What's the chance of two people in this group sharing the same birthday?

Let $n = 2$. What is the probability that two people share the same birthday if there are only **two** people?

Ans: $1/365$

Birthday paradox

Let $\boldsymbol{n = 3}$. What is the probability that two people share the same birthday if there are **three** people?

Let A be the even that at least two people share the same birthday.
Then A^c is the event that all people have distinct birthdays.

$$P(A) = 1 - P(A^c) = 1 - \frac{365 * 364 * 363}{365^3} \approx 0.008$$

prob. of a collision

prob. of no collision

Birthday paradox

In general, if there are n people (with $n < 365$) then the probability that at least two share the same birthday is:

$$P(n) = 1 - \frac{(365)(364)\ldots\left(365-(n-1)\right)}{365^n}$$

$$= 1 - \frac{365!/(365-n)!}{365^n}$$

n	P(n)
5	2.7%
10	11.7%
20	41.1%
23	50.7%
30	70.6%
40	89.1%
50	97.0%
60	99.4%
70	99.9%

Connection to Hash Functions

- Number of people = number of elements in the array
- Days of the year = number of memory locations
- Hash function: $h(person) = birthday$
- Collisions mean that two people share the same birthday.
- People who don't share a birthday with anyone else = isolated elements in a hash function

In general, if there are n elements and k memory locations (with $n \leq k$) then the probability of collision is:

$$P(n,k) = 1 - \frac{k!/(k-n)!}{k^n}$$

Of course, it is trivial to see that $P(n,k) = 1$ if $n > k$

Probability in Hash Functions

Suppose we are hashing n elements into k locations. What is the expected number of isolated elements in a hash function (i.e. number of locations with exactly 1 item)?

Let X be the random variable that counts the number of isolated elements.

Then

$$\mathrm{X} = \sum_{i=1}^{n} X_i$$

$\mathbb{P}(X_i = 1) = \mathbb{P}($ other $n-1$ elements take $k-1$ spots $)$

Where $\mathrm{X_i} = 1$ if $\mathrm{X_i}$ is an isolated element and $\mathrm{X_i} = 0$ otherwise.

$$\mathrm{E(X_i)} = \mathrm{P(X_i} = 1) = \frac{k * (k-1)^{n-1}}{k^n} = \left(\frac{k-1}{k}\right)^{n-1}$$

$= \left(\frac{k-1}{k}\right)^{n-1}$

So,

$$\mathrm{E(X)} = \sum_{i=1}^{n} E(X_i) = n * \left(\frac{k-1}{k}\right)^{n-1}$$

Probability in Hash Functions

Suppose we are hashing n elements into k locations. What is the expected number of empty locations?

Let X be the random variable that counts the number of empty locations.
Then

$$\mathrm{X} = \sum_{i=1}^{k} X_i$$

Where $\mathrm{X_i} = 1$ if $\mathrm{X_i}$ is an empty location and $\mathrm{X_i} = 0$ otherwise.

X_i is empty then we have to place all n elements into the other $k-1$ spots

$$\mathrm{E(X_i)} = \mathrm{P(X_i = 1)} = \frac{(k-1)^n}{k^n} = \left(\frac{k-1}{k}\right)^n$$

So,

$$\mathrm{E(X)} = \sum_{i=1}^{k} E(X_i) = k * \left(\frac{k-1}{k}\right)^{n-1}$$

Announcements

- Final Exam: **Wednesday, December 12, 8:00am - 10:59am** in Price Center Theater.
- You are allowed to bring one page of notes, standard size, both sides
- Solutions to practice exam, old exam, and HW8 will be available **on TritonED this Friday** (12/7) after 4pm
- **HW8 still dues Sunday** 12/9, but it will be graded for completion only.

iclicker points :

- RN only points from lecture
- points from sections will be added on Fri.
- 18 pts = 5%
  17 = 4%
  16 = 3%
  ⋮