CSE 250 Recitation

November 24 - 25: Hash Tables

Discussion: Sets vs Maps

Remember: A hash table is a data structure…it can be used to implement multiple ADTs, like Sets and Maps

How would you implement Sets using a hash table? What about Maps?

- What are the differences?
- What are the runtimes of the main operations?

Come up with some examples of Sets vs Maps.

hashCode vs equals

Remember: Just because two objects map to the same hash code or same hash bucket, does not mean they are equal!

Consider **`BZPair`** in PA3 — we have overridden both the **`hashCode`** and **`equals`** functions so that BZPair can be used as a Key in our hash table

- **`hashCode`** returns an integer used to determine the bucket — two **`BZPairs`** with different birthday/zipcode COULD have the same hash code
- **`equals`** returns true **only if the birthday and zipcode are equal**

Hashing w/Chaining

hash(A) = 636

hash(B) = 712

hash(C) = 459

hash(D) = 12

hash(E) = 157

Exercise

1. Start with a 5-bucket hash table (with chaining) and insert A-E.
   a. What is the load factor?
2. Rehash to an array of size 10.
   a. What is the load factor?
3. Write the pseudocode for lookup, insertion, and removal.

Hashing w/Open Addressing

hash(A) = 636

hash(B) = 712

hash(C) = 459

hash(D) = 12

hash(E) = 157

Exercise

1. Start with a 5-bucket hash table (with open-addressing) and insert A-E
2. Confirm lookup works for all 5 keys
3. Rehash to an array of size 10
4. What if we try to lookup F which hashes to 22?
5. Remove B...confirm lookup still works

Hashing w/Cuckoo Hashing

$h_1(A) = 312$ $h_2(A) = 636$

$h_1(B) = 712$ $h_2(B) = 242$

$h_1(C) = 459$ $h_2(C) = 684$

$h_1(D) = 121$ $h_2(D) = 871$

$h_1(E) = 154$ $h_2(E) = 939$

Exercise

1. Start with a 5-bucket hash table (with Cuckoo Hashing) and insert A-E
2. Rehash as needed…

Cuckoo Hashing Exercise

Imagine we are inserting **A**, **B**, and **C** into a hash table using Cuckoo Hashing…

1. Come up with unique hash values for **A**, **B**, and **C** that would require the hash table to rehash if there are 10 buckets
2. Do the same that would require the hash table to rehash for 20 buckets
3. Can you pick a set of unique hash values that would require the hash table to resize for both 10 **and** 20 buckets, but not 40?

Expected Value

A **random variable** represents a quantity that is dependent on random occurrence

Example: Let **X** be the value rolled on a six-sided die

X is a random variable

- It's value can be 1,2,3,4,5 or 6
- It’s value depends on an event (die landing on the “6” face)
- The event depends on a random experiment (rolling the die)

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Example: Let **X** be the value rolled on a six-sided die

Possible values of **X**: 1,2,3,4,5,6

Probability of each outcome: $\frac{1}{6}$

E[X] $= \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = 3.5$

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Generally:

$$E[X] = \sum_i P_i \cdot X_i$$

Probability of the i^{th} outcome

value of the i^{th} outcome

Exercise

A deck of cards contains 52 cards. 4 aces, 4 of each number 2-10, and 12 face cards (4 jacks, 4 queens, 4 kings).

In Blackjack, number cards are worth their value (ie 2 is worth 2, 3 is worth 3, etc), face cards are worth 10, and aces (for simplicity) are worth 1.

If you draw a single card from a shuffled deck of cards, what is the expected value of that card?

$$E[X] = \sum_i P_i \cdot X_i$$

Exercise

Note: 4/52 = 1/13

If **X** is the value of the drawn card, then

$E[X] = 1/13 \cdot (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10) + 3/13 \cdot 10 = 6.538$