

# The Count Distinct Problem

Steven Rosendahl

November 11, 2015

# Summary

The Problem

The Hash Table

The HyperLogLog

# The Question

# The Question

- ▶ What is the problem?

# The Question

- ▶ What is the problem?
- ▶ Imagine we have a set called  $\mathbb{V}$  that contains a billion elements of the same type.

# The Question

- ▶ What is the problem?
- ▶ Imagine we have a set called  $\mathbb{V}$  that contains a billion elements of the same type.
- ▶ How many *unique* elements are in  $\mathbb{V}$ ?

# Questions

# Questions

1. Pokémon Problem: How many unique Pokémon will a player encounter in a given playthrough of all the games?



# Questions

1. Pokémon Problem: How many unique Pokémon will a player encounter in a given playthrough of all the games?
2. Facebook Problem: How many unique application installs are made a day on Facebook?

# Questions

1. Pokémon Problem: How many unique Pokémon will a player encounter in a given playthrough of all the games?
2. Facebook Problem: How many unique application installs are made a day on Facebook?
3. Twitter Problem: How many unique hashtags are made a day on Twitter?

# Questions

1. Pokémon Problem: How many unique Pokémon will a player encounter in a given playthrough of all the games?
2. Facebook Problem: How many unique application installs are made a day on Facebook?
3. Twitter Problem: How many unique hashtags are made a day on Twitter?

We will use  $\mathbb{S}$  to represent the set of all the data, and  $\mathbb{V}$  to represent the set of unique elements.

# Summary

The Problem

The Hash Table

The HyperLogLog

# Hashing

# Hashing

- ▶ What is *hashing*?

# Hashing

- ▶ What is *hashing*?
- ▶ Applying a function  $h(x)$  to every element in  $\mathbb{S}$ , and storing the result in  $\mathbb{V}$ .

# Hashing

- ▶ What is *hashing*?
- ▶ Applying a function  $h(x)$  to every element in  $\mathbb{S}$ , and storing the result in  $\mathbb{V}$ .
- ▶ Ideally,  $h(x)$  is



# Hashing

- ▶ What is *hashing*?
- ▶ Applying a function  $h(x)$  to every element in  $\mathbb{S}$ , and storing the result in  $\mathbb{V}$ .
- ▶ Ideally,  $h(x)$  is
  1. Onto (surjective)

# Hashing

- ▶ What is *hashing*?
- ▶ Applying a function  $h(x)$  to every element in  $\mathbb{S}$ , and storing the result in  $\mathbb{V}$ .
- ▶ Ideally,  $h(x)$  is
  1. Onto (surjective)
  2. One-to-one (injective)

# Hashing

- ▶ What is *hashing*?
- ▶ Applying a function  $h(x)$  to every element in  $\mathbb{S}$ , and storing the result in  $\mathbb{V}$ .
- ▶ Ideally,  $h(x)$  is
  1. Onto (surjective)
  2. One-to-one (injective)
- ▶ We can ignore the duplicate values in  $\mathbb{V}$ .

# Solving the Pokémon Problem

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value
- ▶ Store the result in  $\mathbb{V}$  if it is not already there.

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value
- ▶ Store the result in  $\mathbb{V}$  if it is not already there.
- ▶ The hash function:
  1. Sum up the ASCII value of each character in a Pokémon's name. Call this  $n$ .



# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value
- ▶ Store the result in  $\mathbb{V}$  if it is not already there.
- ▶ The hash function:
  1. Sum up the ASCII value of each character in a Pokémon's name. Call this  $n$ .
  2. Add  $n$  to the Pokémon's corresponding National Pokédex number. Call this  $m$ .

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value
- ▶ Store the result in  $\mathbb{V}$  if it is not already there.
- ▶ The hash function:
  1. Sum up the ASCII value of each character in a Pokémon's name. Call this  $n$ .
  2. Add  $n$  to the Pokémon's corresponding National Pokédex number. Call this  $m$ .
  3. Find  $m \bmod 721$ .

# Solving the Pokémon Problem

- ▶ How can we solve the Pokémon Problem using a hash?
- ▶ Create a hash function that turns a given Pokémon into a numerical value
- ▶ Store the result in  $\mathbb{V}$  if it is not already there.
- ▶ The hash function:
  1. Sum up the ASCII value of each character in a Pokémon's name. Call this  $n$ .
  2. Add  $n$  to the Pokémon's corresponding National Pokédex number. Call this  $m$ .
  3. Find  $m \bmod 721$ .

# Problems With The Hash Table

# Problems With The Hash Table

- ▶ Memory Intensive
  - ▶ Pokémon problem only dealt with a set  $\mathbb{S}$  of size 6000

# Problems With The Hash Table

- ▶ Memory Intensive
  - ▶ Pokémon problem only dealt with a set  $\mathbb{S}$  of size 6000
  - ▶ Twitter Problem deals with  $\mathbb{S}$  of size 200,000,000.

# Problems With The Hash Table

- ▶ Memory Intensive
  - ▶ Pokémon problem only dealt with a set  $\mathbb{S}$  of size 6000
  - ▶ Twitter Problem deals with  $\mathbb{S}$  of size 200,000,000.
  - ▶ Collisions and collision policies also add to the amount of memory required.

# Summary

The Problem

The Hash Table

The HyperLogLog



# The Algorithm

# The Algorithm

1.