

19.1.2 A second attempt: DataIndexedWordSet

Hashing, Video 2 Intro, Data Indexed English Word Sets



Our `DataIndexedIntegerSet` only allowed for integers, but now we want to insert the `String` `"cat"` into it. We'll call our data structure that can insert strings `DataIndexedEnglishWordSet`. Here's a crazy idea: let's give every string a number. Maybe `"cat"` can be `1`, `"dog"` can be `2` and `"turtle"` can be `3`.

(The way this would work is — if someone wanted to add a `"cat"` to our data structure, we would 'figure out' that the number for `"cat"` is `1`, and then set `present[1]` to be `true`. If someone wanted to ask us if `"cat"` is in our data structure, we would 'figure out' that `"cat"` is `1`, and check if `present[1]` is `true`.)

But then if someone tries to insert the word `"potatocactus"`, we don't know what to do. We need to develop a general strategy so that given a string, we can figure out a number representation for it.

Here are the two main strategies we chose to use:

Strategy 1: Use the first letter.

A simple idea is to just use the first character of any given string to convert it to its number representation. However, if someone tried to insert two words with the same first letter, we have a **collision**, which we deal with using the next strategy.

Strategy 2: Avoid Collisions.

There are 26 unique characters in the English lowercase alphabet. We assign each one a number: $a = 1, b = 2, \dots, z = 26$. Now, we can write any unique lowercase string in **base 26**. (Note that **base 26** simply means that we will use **26** as the multiplier, much like we used **10** and **2** as examples above.)

- “*cat*” = $3 * 26^2 + 1 * 26^1 + 20 * 26^0$

This representation gives a unique integer to every English word containing lowercase letters, much like using base 10 gives a unique representation to every number. We are guaranteed to not have collisions.

[Previous](#)

[19.1.1 A first attempt: DataIndexedIntegerSet](#)

[Next](#)

[19.1.3 A third attempt: DataIndexedStringSet](#)

Last updated 1 year ago

