Whiteboard33

Wednesday, August 19, 2020 8:16 PM

PROBLEM DOMAIN

- Write a function that LEFT JOINs two hashmaps into a single data structure.
- The first parameter is a hashmap that has word strings as keys, and a synonym of the key as values.
- The second parameter is a hashmap that has word strings as keys, and antonyms of the key as values.
- Combine the key and corresponding values (if they exist) into a new data structure according to LEFT JOIN logic.
- LEFT JOIN means all the values in the first hashmap are returned, and if values exist in the "right" hashmap, they are appended to the result row. If no values exist in the right hashmap, then some flavor of NULL should be appended to the result row.
- The returned data structure that holds the results is up to you. It doesn't need to exactly match the output below, so long as it achieves the LEFT JOIN logic.
- Avoid utilizing any of the library methods available to your language.

ALGORITHM

Create a Method called LeftJoin
Create first hash map
Create second hash map
Create third hash map
Create entry point to first hash map
IF second hash map has key - get it
IF value for entry equals first entry
Third hash map gets the key and value and combines them

EDGE CASES

Null values
No built-in methods

BIG O
O(1) - Time
O(1) - space

(synonyms antonyms)
Input<---key1 value1, key2 value2

Output <-- key3, value3

VISUALS

INPUT

Synonym Hashtable:

Key:	Value:
fond	enamored
wrath	anger
diligent	employed
outift	garb
guide	usher
...	...

Antonym Hashtable:

Key:	Value:
fond	averse
wrath	delight
diligent	idle
guide	follow
flow	jam
...	...

OUTPUT

```
[
  ["fond", "enamored", "averse"],
  [ "wrath", "anger", "delight"],
  ["diligent", "employed", "idle"],
  ["outfil", "garb", NULL],
  ["guide", "usher", "follow"]
]
```

PSEUDO CODE

```
Method LeftJoin
Map<String, String> firstMap = new HashMap<>();
Map<String, String> secondMap = new HashMap<>();
Map<String, String> join = new HashMap<>();

for (Entry<String, String> entry : firstMap.entrySet())
   if (secondMap.containsKey(entry.getKey())) {
      String value = secondMap.get(entry.getKey());
      if (value.equals(entry.getValue())) {
         join.put(entry.getKey(), value);
      }
   }
}
```