AnthroChassidus

Samuel Sicklick, Oct. 24 2021

Explanation of Solution Design

In order to determine the lower bound of chassidus types I built a custom UnionFind which would assist in analyzing the interview data to formulate groups of unspecified types. My implementation of UnionFind was based on two HashMaps: one to map each *ID* to a parent or canonical element, and the second to map each canonical element to its quantity of *ID*s. The input arrays of interviewed *ID*s is iterated through, where *union* is called on each pair of elements at index *i*. This call places both elements in the *interviewedMap*, if they are not already entered into it, and sets them at a quantity of 1 in the *chassidusQuantMap*. The smaller of the two trees, as determined by what quantity their canonical elements map to, is then set to be the child of the larger tree's canonical element, and the *chassidusQuantMap* is adjusted accordingly. The calls to *find*, which provide the canonical elements for the *union*, function via path compression in adjustments to the *interviewedMap* tree. The inputted *int* is set to some variable *i*, which is checked repeatedly so long as it does not equal the canonical element. Each repeated check on this variable *i*, where it is not the canonical element, is followed by *i*'s parent being set to its grandparent and then *i* being set to its parent. This effectively shortens the tree as it makes each lower leaf closer to the canonical element, and eventually will return the canonical element itself when *i* reaches that value.

Since all calls to *put* and *get* in *HashMap* run in constant time, as per the javadocs on the JDK's implementation of it, the *constructor* of *AnthroChassidus* is O(n). Meaning, the constructor runs O(1) operations in each call to *union* (which includes calls to *find* that are also constant) on the iterating index *i* in both input arrays, thereby rendering the time complexity of the *constructor* O(n). The last line in the *constructor* calculates the lower bound of chassidus types by subtracting from the population size *n* the number of individual *ID*s which were interviewed, and then adds back the number of groups of chassidus types formed by those interviewed *ID*s. This value is stored as an instance variable which is returned by calling *getLowerBoundOnChassidusTypes*, thus rendering it O(1). Lastly, the *nShareSameChassidus* method gets a root via *find* on the inputted *ID*, which is then used to get the value of the returned canonical element's chassidus type group's size from the *chassidusQuantMap*. Such a call also only involves constant time operations, thus establishing the method as running in O(1).