Oliver Moseler

Performance test 4

The method *contains* fo class *com.google.common.collect.RegularImmutableSet*, which is a hash table internally has a bad runtime on a specific input. Comparing it to the method *contains* of the class *HashSet*, which has way better runtime on the same input, shows that it can be faster.

Task

Find the reason for the bad runtime! It's actually no performance bug!

Problem is

The hash table does linear probing in case a hash collision occurs. The hash function or more precisely the *Hash.smear()* method of class *Hashing* is really bad for the provided input. The fact that a consecutive sequence of values gets inserted ends up in the problem that there are no gaps (indexes where no value is stored) in the hash table and so half of the hash table consecutively stores values and the other half is empty. Thus calling contains turns into a linear search. Searching for values that are assuredly not stored in the hash table provokes the method contains to iterate through all values till it finds a gap. But a gap first occurs after the last stored value as explained above!

Solution

Make a change in the method *smear()* in class Hashing.

Hints

1. The method *contains* does linear probing.
2. The values within the hash table are stored in a consecutive sequence.
3. Hashing with linear probing depends on the quality of the hashing function.