

Question 2. Double Hashing

The **ProbeHashMap** class outlined in Chapter 10.2 of your textbook uses linear probing. Here we will extend this class to use double hashing as outlined in Slides 34-36 of the Maps and Hash Tables lecture. You are provided with a functional implementation of **ProbeHashMap**, enhanced to ensure that the capacity of the map is always a prime number. You are also given a partial implementation of **DoubleProbeHashMap**, which will extend **ProbeHashMap** to use double hashing. You must implement three methods: **resize(newCap)**, which resizes and rehashes the map, **findSlot(h1, k)**, which finds the location of the entry with key k or, if such an entry does not exist, returns the location where a new entry with key k should be stored, and **secondaryHashValue(k)**, which computes the secondary hash value for key k.

To test your code, you can use the **testDoubleProbeHashMap** program, which reads a comma-separated data file **marathon2017.csv** containing records for all competitors in the 2017 Toronto Marathon and stores them in a map where the key is a concatenation of the runner's name and Age/Gender Category (Surname + Given Name + Category). **marathon2017.csv** should be stored in the top-level directory of your Assignment 3 project.

testDoubleProbeHashMap is currently configured to use **ProbeHashMap** - you can modify it to use **DoubleHashMap** once you have it implemented.

Note that **testDoubleProbeHashMap** reports the total number of probes before exiting. Which hashing method seems more efficient? (You do not have to submit your answer to this question - this is for your own interest.)

Here is the [code package](#) and [data file](#). Have fun!