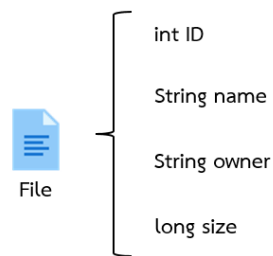


File Bucket

You are implementing a web application, and your backend requires file storage to store user's uploaded files, but you don't want to use any cloud services (e.g. AWS S3 or Firebase Storage). So, you implement a file bucket which does not require sorting using an open addressing hash table with double hashing probing.

In the open addressing hash table, the array inside the table will store file objects, which include ID, name, owner name, and file size. View the class **File** for more information.



In class **FileBucketDoubleHashing**, implement 5 methods:

1. `public int hash(int id)`

This method returns hashed value of the ID of the file. The hash function is

$$h(x) = x \% \text{size of array in the table}$$

2. `public int hash2(int id)`

This method returns hashed value of the ID of the file from second hash function. The second hash function is

$$h(x) = \begin{cases} x \% 5 & \text{when } x \text{ is even integer} \\ x \% 4 & \text{when } x \text{ is odd integer} \end{cases}$$

3. `public void add(File file) throws Exception`

This method will add the file object into the hash table. The hash values used in the method should come from `hash(id)` and `hash2(id)`. The flow of this method is similar to normal open addressing hash table. Note that 2 file objects are equivalent when ID of the files are equal.

4. `public void rehash() throws Exception`

This method will rehash the hash table (due to exceeding load factor). The flow of this method is similar to normal open addressing hash table.

5. `public void remove(int id)`

This method will remove the file with specific ID. The flow of this method is similar to normal open addressing hash table. Also, don't forget to set the flag of the table when the file is deleted. You can use protected static final File ***DELETED*** to set the flag (view more in class **FileBucketOpenAddressing**).

Note that 2 file objects are equivalent when ID of the files are equal. Also, the normal open addressing hash table with double hashing probing is provided in the directory.

Guarantee that there is no file with the same ID when inserting in the test cases.