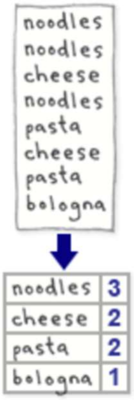
PA - 2
Hash Tables

Description:

For this programming assignment, you will implement **three different versions of a hash table** and study their performance. You will implement (1) Chaining, (2) Linear Probing and (3) Double Hashing.

Your program will be a simple **word-counting application**. It reads a text file and creates a hash table, mapping each word to its number of occurrences. Your program should take in and process a .txt file as input (example dictionary.txt). The file **dictionary.txt** has one million words in which there are about 58,000 unique words, each occurring a different number of times. Once you have inserted all the words into a hash table, your program should output the number of occurrences for all given strings in the .txt file.



noodles
noodles
cheese
noodles
pasta
cheese
pasta
bologna

noodles	3
cheese	2
pasta	2
bologna	1

Your program should use an array-based implementation for the hash table. The program should also implement a hash function that takes the ASCII values of each character within a word, multiplies them by a positive constant, adds them together, then compresses them with the modulo function (pseudocode below).

```
hash = 0
n = len(s) # s is the given word
for i in range(n):
    # g is a positive constant chosen by you
    hash = g * hash + ASCII value of char at position i in the word

index = hash % capacity # capacity is total size allocated for hash table
```

For the Double Hashing portion, you should review and consider course materials and literature regarding double hashing and implement an efficient secondary hash function. Be sure to give this some critical thought. Add a description of this implementation as markdown in your script.

Skeleton code is provided. Please do not change function names or class names. Additional classes can also be added.

Coding Portion (30 Points):

- Create the three implementations of the different hash tables.
- Be sure to test the correctness of your algorithms and implementations.
- Submit the .py file and additional helper files you create (if any) along with the text files which contain your hash table outputs.
- Your code will be graded based on whether or not it run, produces correct output on test inputs, and your coding style.
- Required functionality:
  - Your classes should take in a file name and then populate the hash-tables appropriately.
  - Finally, it should print the contents of the hash tables into three separate files titled "Probing.txt", "DoubleHashing.txt", "Chaining.txt",
  - For grading purposes, the generated files will be compared against files containing the expected output.
- You should also insert markdown text describing the strategy and data structure used for each implementation in the code itself as a text cell.
- Run some exploratory performance tests and briefly describe *which implementation performs the best. Is there an implementation that performs better for all capacities, or are certain implementations more efficient at different capacities?*

Submission:

Once you have completed the assignment, you should upload the python script and output files to the **PA-2** assignment portal on Brightspace. The python script file should be named **PA2.py**.