Data Structures - Lab Worksheet

Dictionaries & Hashtables

Start by downloading the provided coding canvas for dictionaries and hashtables.

Question 1 - Dictionaries

Implement the incomplete methods `put(k,v)`, `get(k)->v`, and `remove(k)->v` in class `DictionaryAsDoubleList`

Question 2 - Hashing and collisions

2.1. Consider that *collisions are handled by **chaining***.
Draw the 11-entry hash table that results from using function h(i)=(3i+5)%11 to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5.

Answer:

index	0	1	2	3	4	5	6	7	8	9	10
	13	94				44			12	16	20
		39				88			23	5	
						11					

2.2. Consider that *collisions are handled by **linear probing***.
Draw the 11-entry hash table that results from using function h(i)=(3i+5)%11 to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5.

Answer:

index	0	1	2	3	4	5	6	7	8	9	10
	13	94	39	16	5	44	88	11	12	23	20

2.3. Consider that *collisions are handled by **quadratic probing***.
Draw the 11-entry hash table that results from using function h(i)=(3i+5)%11 to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5.

Answer:

index	0	1	2	3	4	5	6	7	8	9	10
	13	94	39	11		44	88	16	12	23	20

The last value 5 has a collision with 23 at index 9, so by quadratic probing, we wish to find an integer j such that 9 + j^2 ≡ 4 (mod 11). However, there does not exist positive integer j such that j^2 ≡ 6 (mod 11). This can be proved by listing j from 1 to 11 (one circle) and none of them satisfies j^2 ≡ 6 (mod 11). Hence 5 can not be put into the hash table by quadratic probing, we therefore need to resize the table.

Question 3 - Hashtable

Implement the incomplete methods `put(k,v)`, `get(k)->v`, and `remove(k)->v` in class `ChainHashtable`

Question 4 - Solving problems with map structures

Write a program count_words.py that reads every word in file `count_words.txt`, and displays the word that occurs the most frequently. Your program should work equally well with a `DictionaryAsDoubleList` and a `ChainHashtable`