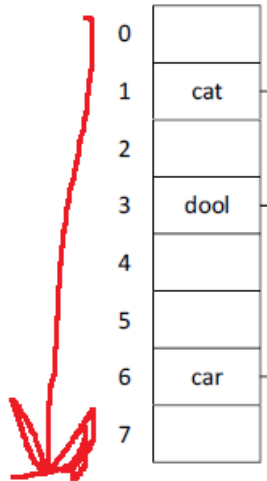


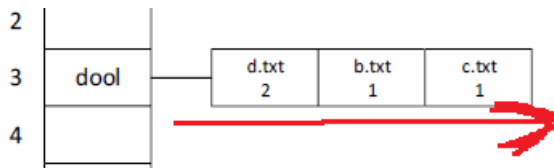
Some significant functions:

private Node locate(K key) : it walk through hashmap and when it found a node whose .getKey() value equal key then it return this not otherwise return null.



private Node locate3(int index, V value) :

If it searches for the value given in the index node of the hashtable, it sends that node, if it can't find it, it sends null



public int whichLetter(char c) : finds the order of the given letter in the alphabet

public int getNextPrime(int x): find the prime number that smallest bigger than x

public static void searchWords(HashedDictionary hashedDictionary): Searches for words in "search.txt" one by one. then the average minimum and maximum values are printed on the screen.

Load Factor	Hash Function	Collision Handling	Collision Count	Indexing Time	Avg. Search Time	Min. Search Time	Max. Search Time
$\alpha=50\%$	SSF	LP	482694	57.1287424	3.013359999999991E-5	2.14E-5	0.0017617
$\alpha=50\%$	SSF	DH	471168	33.3536919	3.12516999999999496E-5	2.26E-5	0.0017734
$\alpha=50\%$	PAF	LP	308114	35.7903404	5.4704100000000005E-5	4.61E-5	0.0019237
$\alpha=50\%$	PAF	DH	306501	35.603574	9.6795799999999986E-5	8.84E-5	0.0020546
$\alpha=80\%$	SSF	LP	482694	54.6922419	3.182100000000021E-5	2.13E-5	0.0019174
$\alpha=80\%$	SSF	DH	471168	30.2468189	3.110290000000003E-5	2.31E-5	0.0018726
$\alpha=80\%$	PAF	LP	308114	35.2769973	5.7643899999999924E-5	4.72E-5	0.0019425
$\alpha=80\%$	PAF	DH	306501	34.6686605	9.710279999999993E-5	8.84E-5	0.0020662