

## Step 1

Num test # of probes: 24

Element		22	44			43	27	7		89	30						85	64			
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

## Step 3

String test # of probes: 19

Element		CK		CM	BD			H					MOB						W	A	X	Y	
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

## Step 4

I think most collisions would be with Huckleberry because all of the words in the dictionary are unique, whereas there are many duplicates in Huckleberry.

Setting the capacity to be twice as large is important because with Quadratic probing, we could quickly get indexes that are out of bounds of our initial, normal sized, array. If we expect many collisions, we should prepare in advance to make space for probing.

The results don't match my predictions. Huckleberry has much less collisions than both dictionary HashTables. This is because every duplicate word in Huckleberry only adds 1 probe, then returns out of the insert function. This can also be seen in the average # of probes for Huckleberry being closer to 1.0 than the others. If there ever is a collision in Huckleberry, it's typically a duplicate, so instead of finding a new hash value and looping back, it just adds 1 probe then kicks out.

Ordered Dictionary # probes: 65509

Huckleberry # probes: 45610

Unordered Dictionary # probes: 65608

Ordered Dictionary average # probes: 1.442866

Huckleberry average # probes: 1.0045813

Unordered Dictionary average # probes: 1.4450464