

## Assignment 5

Changrong chen 001276880

This assignment is to check two situation if they are the same as given equation. These two situation are 1. average number of hashes before the first collision in encountered 2. average number of hashes before all slots are filled.

The capacity of slots are assigned as those numbers:

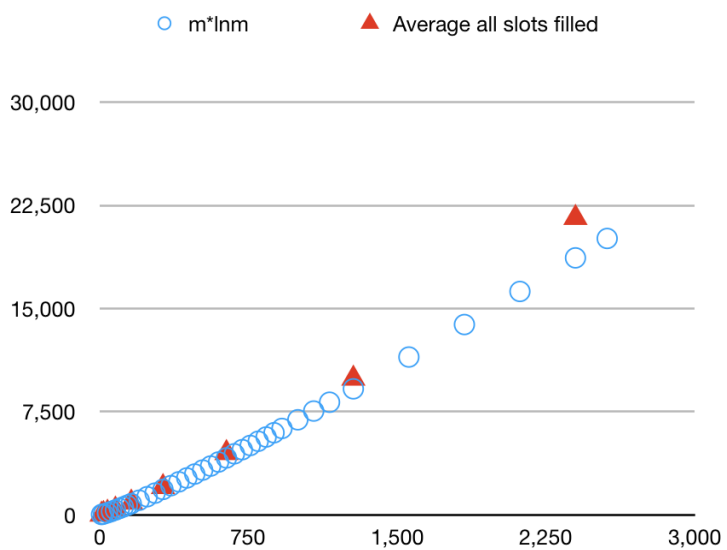
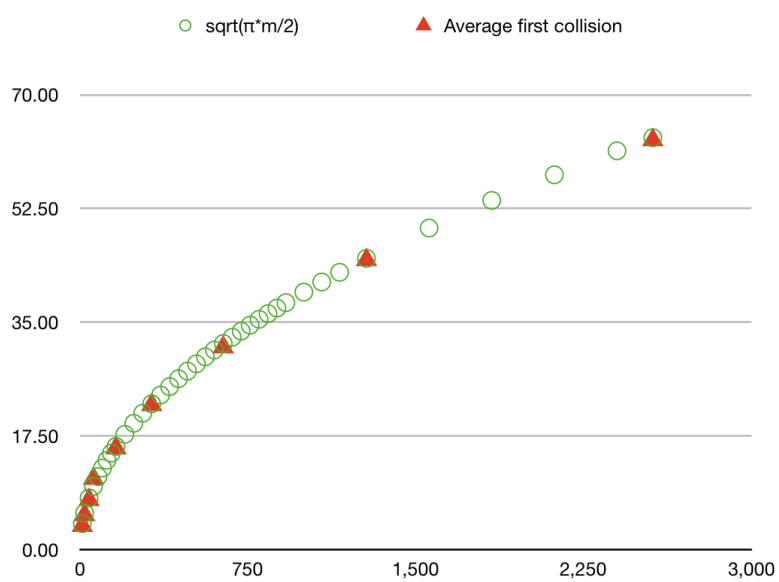
10, 20, 40, 80, 160, 320, 640, 1280, 2560

And every experiment will run 5000 times to get a average number to keep accurate.

The following is running results and data:

m	Average first collision	$\sqrt{\pi \cdot m / 2}$		Average all slots filled	$m \ln m$	
10	3.673	3.960	0.928	29.451	23.0258509299405	1.279
20	5.3292	5.6000	0.952	71.9364	59.9146454710798	1.2006
40	7.6548	7.9300	0.965	170.3116	147.555178164557	1.1542
60		9.7100			245.660673733326	
80	10.929	11.2100	0.975	398.7358	350.56213077391	1.1374
100		12.5300			460.517018598809	
120		13.7300			574.499009133845	
140		14.8300			691.829939165303	
160	15.6142	15.850	0.985	904.692	812.027810437412	1.114
200		17.720			1059.66347330961	
240		19.420			1315.35334160208	
280		20.970			1577.74108888739	
320	22.284	22.4200	0.994	2035.0706	1845.86271865401	1.1025
360		23.7800			2118.99745132206	
400		25.0700			2396.58581884319	
440		26.2900			2678.18087984141	
480		27.4600			2963.41732987293	
520		28.5800			3251.99098201925	
560		29.6600			3543.64459888835	
600		30.7000			3838.15779312969	
640	31.1704	31.710	0.983	4506.724	4135.33963286638	1.090
680		32.674			4435.0231027557	
720		33.620			4737.06087264727	
760		34.543			5041.32200929309	
800		35.440			5347.68938213434	
840		36.315			5656.05758914338	
880		37.170			5966.33127857558	
920		38.000			6278.42377643964	
1000		39.623			6907.75527898214	
1080		41.178			7543.49362572773	
1160		42.676			8185.16332955648	
1280	44.6178	44.8285	0.995	9914.9616	9157.90765684949	1.0827
1560		49.4894			11469.80811638	
1840		53.7475			13832.2383651096	
2120		57.6923			16237.443299452	
2400		61.3840			18679.7376392065	
2560	63.0752	63.4123	0.995	21601.1298	20090.27	1.0752

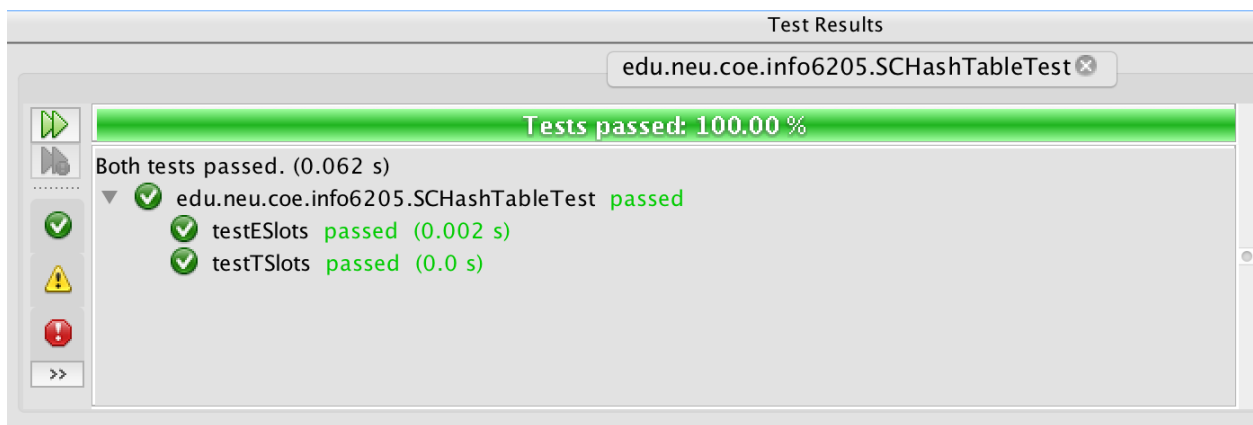
/\*  
3.673,29.451  
5.3292,71.9364  
7.6548,170.3116  
10.929,398.7358  
15.6142,904.692  
22.284,2035.0706  
31.1704,4506.724  
44.6178,9914.9616  
63.0752,21601.1298  
\*/



From the experiment we can get these result:

- The average number of hashes before the first collision encountered follows is  $\sqrt{\pi \cdot m/2}$
- The average number of hashes before all slots are filled is closer to  $(m \cdot \ln m)$ , the more slots, the more trends to equal.(when slots is 10, the ration is 1.279; when slots is 2560, the ration is 1.0752)

Unit test Result & Code:



```

/**
 * Test of eSlots method, of class SHashTable.
 */
@Test
public void testESlots() {

    SHashTable<Double, Integer> ht = new SHashTable<>(5);
    Random r = new Random();
    double x = r.nextDouble();
    ht.put(x, 1);
    Assert.assertTrue(!ht.tSlots());

    SHashTable<Double, Integer> ht1 = new SHashTable<>(5);
    for(int i=0; i<6; i++){
        Random r1 = new Random();
        double x1 = r1.nextDouble();
        ht1.put(x1, 1);
    }
    Assert.assertTrue(ht.eSlots());
}

/**
 * Test of tSlots method, of class SHashTable.
 */
@Test
public void testTSlots() {

    SHashTable<Double, Integer> ht2 = new SHashTable<>(5);
    Random r = new Random();
    double x = r.nextDouble();
    ht2.put(x, 1);
    Assert.assertTrue(ht2.eSlots());

    SHashTable<Double, Integer> ht3 = new SHashTable<>(5);
    for(int i=0; i<4; i++){
        Random r2 = new Random();
        double x2 = r2.nextDouble();
        ht3.put(x2, 1);
    }
    Assert.assertTrue(ht3.eSlots());
}

```