Using the skeleton app provided in the ***/Interesting Stuff/Hash Tables – Chaining/***folder, complete the implementation of a *Separate chaining with linked lists* collision resolution hash table.

Use the provided input files (100.data, 1000.data etc.) to feed your hash table.

Requirements:

* Implement all the operations as described in the skeleton
* At the end, after every element has been inserted, compute the standard deviation of all the buckets’ sizes. (see [here](http://en.wikipedia.org/wiki/Standard_deviation#Basic_examples) an example of how to compute this) – also, read up on what standard deviation is and try to figure out what a low/high value of such a measure would mean in our case.
* Try things out with a bad hash function first (a H0 – think about the worst hash function which could still somehow distribute entries in more than 1 bucket)
* tabel is for \_100 case

|  |  |  |
| --- | --- | --- |
| **Hash Function (char \* c, int i);** | **ISF \*** | **σ (standard deviation)** |
| H0 | .10 | 4.76 |
| H0 | .20 | 9.76 |
| H0 | .35 | 17.38 |
| H0 | .50 | 23.36 |
| H0 | .75 | 37.11 |
| H0 | .85 | 37.61 |
| H1 | .10 | 3.99 |
| H1 | .20 | 9.59 |
| H1 | .35 | 15.81 |
| H1 | .50 | 23.79 |
| H1 | .75 | 35.04 |
| H1 | .85 | 40.29 |
| H2 | .10 | 3.74 |
| H2 | .20 | 10.24 |
| H2 | .35 | 15.36 |
| H2 | .50 | 24.64 |
| H2 | .75 | 39.39 |
| H2 | .85 | 46.64 |
| H3 | .10 | 3.97 |
| H3 | .20 | 11.17 |
| H3 | .35 | 8.05 |
| H3 | .50 | 11.97 |
| H3 | .75 | 11.97 |
| H3 | .85 | 71.97 |

\* ISF = Initial Size Factor

Have other combinations in mind? Feel free to fill-up the table with more tries to see if any interesting results come up!

H0: bad hash function!

**i=content[1];**

H1:

***int hashFunction(char \* content, int i)***

***{***

***int length = strlen(content);***

***int k, sum;***

***for (sum=0, k=0; k < length; k++)***

***{***

***sum += content[k];***

***}***

***return sum % size;***

***}***

H2 : int length=strlen(content);

int k;

for (k=0;k<length;k++)

{

i=i\*332+content[k];

i=i%size;

}

H3 : int length=strlen(content);

int k;

for (k=0;k<length;k++)

{

i+=content[k]\*2654435761 % 2^32;

}

Deadlines:

30411 – 18.05.2015 (before 12:00 if you want review and before 23:59 if you want a grade)  
30414 – 19.05.2015 (before 12:00 if you want review and before 23:59 if you want a grade)