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)

|  |  |  |  |
| --- | --- | --- | --- |
| **Hash Function (char \* c);** | **Input\_FILE** | **ISF \*** | **σ (standard deviation)** |
| H0 | \_100 | .10 | 26.6833 |
| H0 | \_100 | .20 | 18.8680 |
| H0 | \_500 | .35 | 32.1559 |
| H0 | \_500 | .50 | 26.8328 |
| H0 | \_1000 | .75 | 31.0161 |
| H0 | \_1000 | .85 | 29.1376 |
| H1 | \_100 | .10 | 9.3274 |
| H1 | \_100 | .20 | 4.2426 |
| H1 | \_500 | .35 | 2.2361 |
| H1 | \_500 | .50 | 1.4142 |
| H1 | \_1000 | .75 | 1.7321 |
| H1 | \_1000 | .85 | 1.7321 |
| H2 | \_100 | .10 | 9.5394 |
| H2 | \_100 | .20 | 4.5826 |
| H2 | \_1000 | .35 | 2.4495 |
| H2 | \_1000 | .50 | 1.4142 |
| H2 | \_25000 | .75 | 1.0000 |
| H2 | \_25000 | .85 | 1.0000 |
| H3 | \_2500 | .10 | 9.5394 |
| H3 | \_2500 | .20 | 4.5826 |
| H3 | \_1000 | .35 | 2.4495 |
| H3 | \_1000 | .50 | 1.4142 |
| H3 | \_500 | .75 | 1.0000 |
| H3 | \_500 | .85 | 1.0000 |

\* 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!

H1:

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

***{***

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

***int k, sum;***

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

***{***

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

***}***

***return sum % size;***

***}***

H2 – H3 🡪 your choices!

* Try to improve each time

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)