Hash Table

Khalid Hourani

April 14, 2017

1. We see that the worst-case search-time of such a hash occurs when the n integer keys are congruent mod m. In this case, all n keys collide and have the same index. Thus, a search through these elements is equivalent to a search through a linked list, which has $O(n)$ worst-case performance.

   For example, take $m = 7$, $n = 10$, with the following keys:

   $$\{52, -32, -53, 45, -4, 3, -67, -11, 59, 10\}$$

   All of these keys have a hash-value of 3, since they are all congruent to 3 mod 7. Then our hash-table is:

Hash	Key									
0										
1										
2										
3	52	-32	-53	45	-4	3	-67	-11	59	10
4										
5										
6										

   And, in the worst-case scenario of searching for an element not in the list with hash-value 3, e.g. an element with a key of -46, we must search through the entire linked list

   $$52 \to -32 \to -53 \to 45 \to -4 \to 3 \to -67 \to -11 \to 59 \to 10$$

   to see that the element is not in our list.

2. For time-critical applications, this hash is not ideal, since the worst-case performance time is $O(n)$. Instead, something with a faster worst-case performance is preferable.