Linear Hashing

笔记本:	CS 562		
创建时间:	2/15/2020 3:26 PM	**更新时间:**	2/15/2020 5:00 PM
作者:	tanziqi1756@outlook.com		
URL:	https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E6%95%A3%E5%88%97		

Linear Hashing

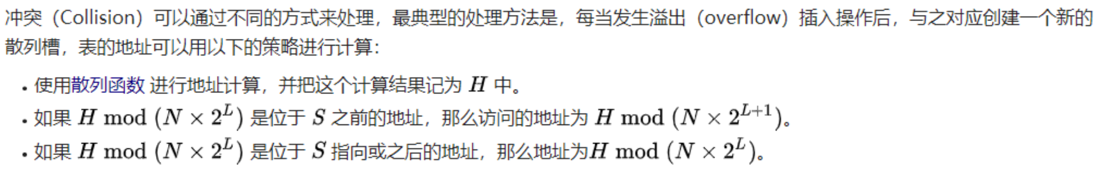
线性散列是由Witold Litwin（1980）发明并被Paul Larson推广的一种动态散列（dynamic hash）算法。线性散列表的每次扩张仅增加一个槽（slot、bucket）， 频繁的单槽扩张可以非常有效控制的冲突链的长度，从而哈希表扩展的代价摊还在每一次插入操作中。 因此非常适合用于交互式应用程序。

Initilization

- N: Capacity of a page(bucket) (Assume N=4);
- L: the current number of the pages;
- Hash Function: Assume hi(x) = h(x) mod 2^i * N;
- S pointer: pointer the next bucket that will be split;
  %.* Split policy: split whenever an overflow page is created. Or split when the utilization of the space (all buckets) has exceeded

insertion

Query



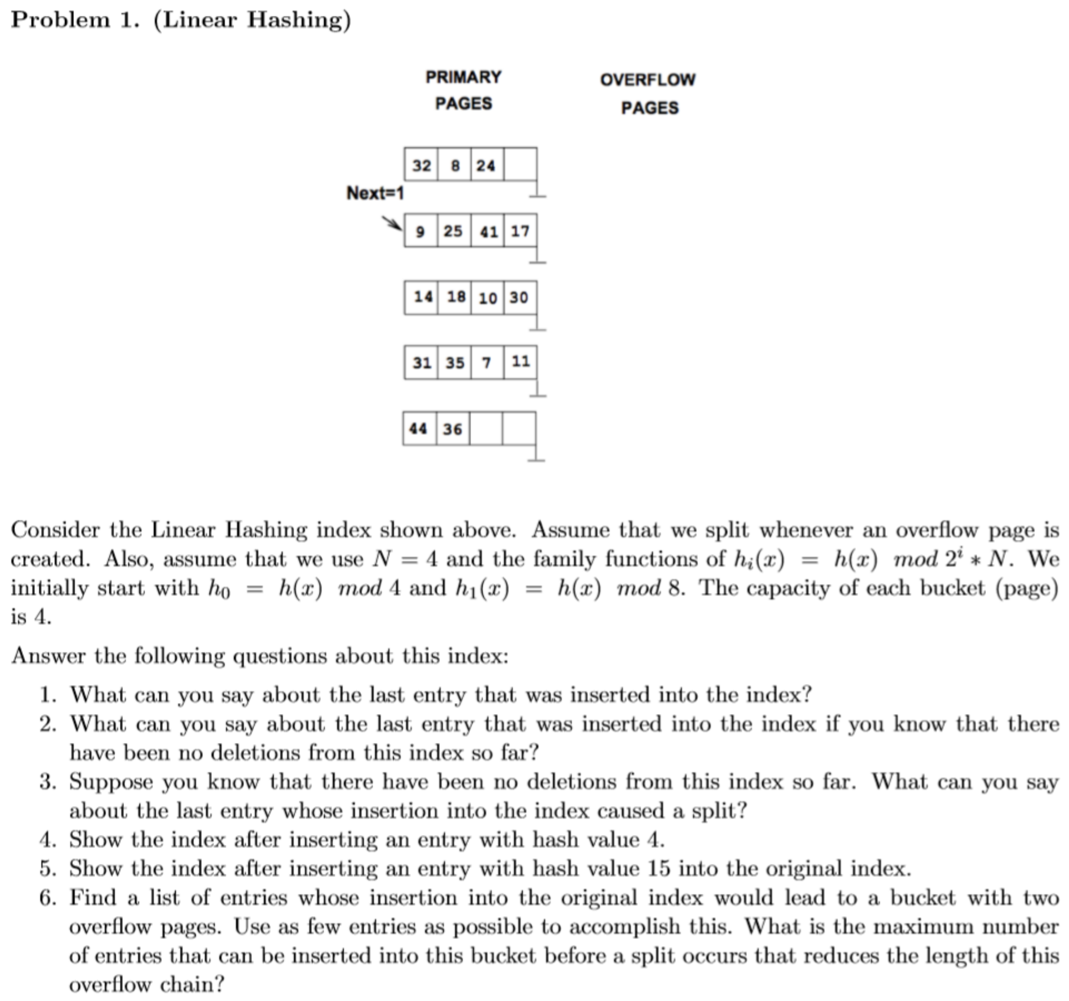
冲突（Collision）可以通过不同的方式来处理，最典型的处理方法是，每当发生溢出（overflow）插入操作后，与之对应创建一个新的散列槽，表的地址可以用以下的策略进行计算：

- 使用散列函数 进行地址计算，并把这个计算结果记为 H 中。
- 如果 $H \bmod (N \times 2^L)$ 是位于 S 之前的地址，那么访问的地址为 $H \bmod (N \times 2^{L+1})$。
- 如果 $H \bmod (N \times 2^L)$ 是位于 S 指向或之后的地址，那么地址为$H \bmod (N \times 2^L)$。

Overflow and Split

1. 在散列表的末尾分配一个新的散列槽。
2. S所指向的bucket的entry进行重新分配。
3. S自增，如果S>2^i, S = 0.

Example



Problem 1. (Linear Hashing)

PRIMARY
PAGES
OVERFLOW
PAGES
Next=1
32 8 24
9 25 41 17
14 18 10 30
31 35 7 11
44 36

Consider the Linear Hashing index shown above. Assume that we split whenever an overflow page is created. Also, assume that we use $N = 4$ and the family functions of $h_i(x) = h(x) \ mod \ 2^i * N$. We initially start with $h_0 = h(x) \ mod \ 4$ and $h_1(x) = h(x) \ mod \ 8$. The capacity of each bucket (page) is 4.

Answer the following questions about this index:

1. What can you say about the last entry that was inserted into the index?
2. What can you say about the last entry that was inserted into the index if you know that there have been no deletions from this index so far?
3. Suppose you know that there have been no deletions from this index so far. What can you say about the last entry whose insertion into the index caused a split?
4. Show the index after inserting an entry with hash value 4.
5. Show the index after inserting an entry with hash value 15 into the original index.
6. Find a list of entries whose insertion into the original index would lead to a bucket with two overflow pages. Use as few entries as possible to accomplish this. What is the maximum number of entries that can be inserted into this bucket before a split occurs that reduces the length of this overflow chain?