Stevens Institute of Technology

Department of Computer Science

CS590: ALGORITHMS

Homework Assignment 5

Submitted by:
Hadia Hameed
CWID: 10440803

Yijia Tan
CWID: 10427079

Submitted to:
Prof. Iraklis
Tsekourakis

November 9, 2018

1 Problem I

We wish to implement a dictionary by using direct addressing on a huge array. At the start, the array entries may contain garbage, and initializing the entire array is impractical because of its size. Describe a scheme for implementing a direct-address dictionary on a huge array. Each stored object should use *O(1)* space; the operations SEARCH, INSERT, and DELETE should take *O(1)* time each; and the initialization of the data structure should take *O(1)* time.

(Hint:*Use an additional stack, whose size is the number of keys actually stored in the dictionary, to help determine whether a given entry in the huge array is valid or not.*)

1.1 Inserting a value k into array A:

In order to store a key k in the array A, we will use a struct called *obj* having the following attributes:

- ***array_key*:** Value in the dictionary that needs to be inserted.
- ***ptr*:** Pointer to the node of the stack.

1.2 Implementing a stack S:

The stack will consist of nodes. Each nodes will have three attributes:

- ***stack_key*:** Pointer to the slot in array A that's pointing to the current node of stack S.
- ***next_ptr*:** Pointer to the node next to the current node in stack S.
- ***prv_ptr*:** Pointer to the node previous to the current node in stack S.

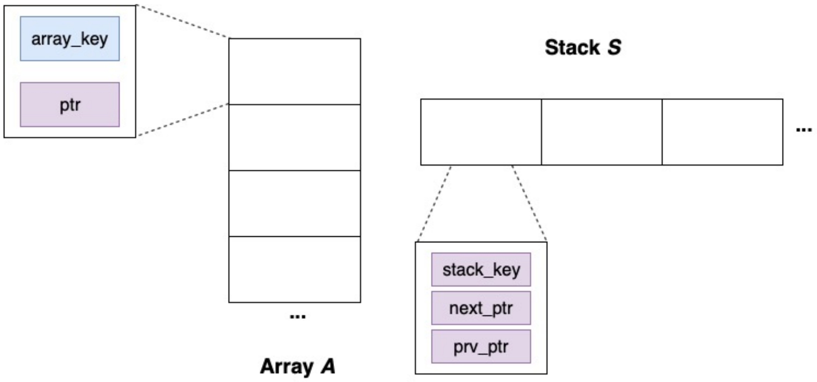


Figure 1: Properties of objects within Array A and nodes within Stack S

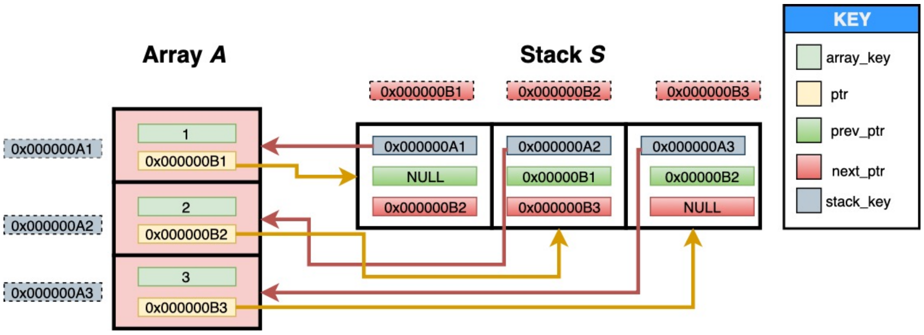


Figure 2: Example of storing dictionary values keys [1,2,3] in Array A.

1.3 Searching a value *k* in array *A*:

Go to the k^{th} slot of A. Get the value of the *ptr* in the corresponding slot of A. This is the address of some node of Stack S. Go to that address. Get the value of the *stack_key* property of that node. If it is equal to the address the slot of A containing k, then return true, otherwise return false.

1.4 Deleting a value *k* in array *A*:

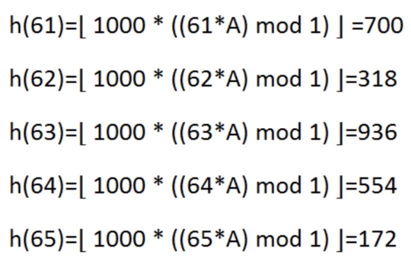
Go to the k^{th} slot of A. Get the value of the *ptr* in the corresponding slot of A. This is the address of some node of Stack S. Go to that address. Set the value of the *stack_key* property of that node to NULL.

2 Problem II:

Consider a hash table of size m = 1000 and a corresponding hash function:

$$h(k) = \lfloor m(kAmod1) \rfloor, A = (\sqrt{5} - 1)/2$$

Compute the locations to which the keys 61, 62, 63, 64, and 65 are mapped.



h(61)=⌊ 1000 * ((61*A) mod 1) ⌋ =700

h(62)=⌊ 1000 * ((62*A) mod 1) ⌋=318

h(63)=⌊ 1000 * ((63*A) mod 1) ⌋=936

h(64)=⌊ 1000 * ((64*A) mod 1) ⌋=554

h(65)=⌊ 1000 * ((65*A) mod 1) ⌋=172