

Homework Assignment #2

Due: January 24, 2025 at 5:00 p.m.

1. Consider the problem of implementing a VECTOR that stores a sequence of integers $x_1, \dots, x_n$ and supports the following operations.

- READ(i) returns x_i .
- WRITE(i, v) changes x_i to v .
- INCSize(v) appends v to the end of the sequence (increasing its length by 1).

Assume that the sequence initially has length 1.

We represent the sequence using two pointers A_1 and A_2 to arrays of size n_1 and n_2 , respectively, and an index $last$ that represents the last location in the second array that is currently in use. We use the notation $*A_1$ to be the array that A_1 points to, so $*A_1[i]$ denotes the i th element of the first array.¹ The elements of the sequence are the elements of the first array followed by the first $last$ elements of the second array. Initially, $n_1 = n_2 = 1$.

```

1: READ(int i) : int                                ▷ Precondition:  $1 \leq i \leq n$ 
2:   | if  $i \leq n_1$  then return  $*A_1[i]$ 
3:   | else return  $*A_2[i - n_1]$ 

4: WRITE(int i, int v)                              ▷ Precondition:  $1 \leq i \leq n$ 
5:   | if  $i \leq n_1$  then  $*A_1[i] \leftarrow v$ 
6:   | else return  $*A_2[i - n_1] \leftarrow v$ 

7: INCSize(int v)
8:   | if  $last = n_2$  then                            ▷ data structure is full; rebuild it
9:   |   allocate new arrays  $B_1$  of size  $n_1 + n_2$  and  $B_2$  of size  $n_1 + 2n_2$ 
10:  |   copy all elements of  $*A_1[1..n_1]$  and  $*A_2[1..n_2]$  into  $B_1$  (in this order)
11:  |   change pointers  $A_1$  and  $A_2$  to point to  $B_1$  and  $B_2$ , respectively
12:  |    $last \leftarrow 0$ 
13:  |    $last \leftarrow last + 1$ 
14:  |    $*A_2[last] \leftarrow v$ 

```

- [1] (a) Draw a picture of the data structure after performing 5 INCSize(42) operations.
- [3] (b) Find a constant c such that $n_2 \leq c \cdot n_1$ always holds. Prove your answer is correct.
- [5] (c) Give good upper bounds on the amortized cost of the operations. Prove your answer is correct.
- [1] (d) Show that when the sequence has n elements, the total space used by the two arrays of the data structure is $O(n)$.

¹In C, array variables actually store the address of the first element of the array, so you could use $*(A_1 + i - 1)$ to access the i th element of the array A_1 .