Foundation of Algorithms

605.421.83

Programming Assignment 1

Name: Guan Yue Wang

ID: gwang39

E-mail: gwang39@jhu.edu

Phone: +16479888206

Date: Oct. 2nd, 2017

1.

(a) [50 points] Given this data structure, prove that the data structure yields O(lg n) time for searching but O($\sqrt{n}$) amortized time for adding an element

Searching - O (lg n)

Since both longer and shorter array are always sorted, binary search takes advantage of this sorted structure and runs O(lg n) time to find an element. To be specific, binary search always compares target value with the mid element of the array first, if the it is not equal, it compares again with the mid value of the half of the interval potentially would have such element. This process goes on until either the value is found or entire array is checked and the value doesn't exist. Consequently, Binary search has O(lg n) with the worst case making lg n comparisons. If we assume the array size of longer array is n and shorter array is $\sqrt{n}$, the search time for the longer array is O (lg n) and for shorter array is O (lg $\sqrt{n}$). Combing both together, this essentially leads to O (lg n).

Adding - is O ($\sqrt{n}$)

To begin with, we always try insert elements into the shorter array first, this leads to O ($\sqrt{n}$) as the worst case happens when the value is smaller than all values in the element and we have to move all elements inside. Additional to this, for every $\sqrt{n}$ insertion, merging and re-allocating need O(n) needs to happen. Therefore, cost O(n) is linked to every$\sqrt{n}$ insertion. As a result, the amortized cost of O(n/$\sqrt{n}$) = O ($\sqrt{n}$)

(b) [50 points] Implement the above data structure and insert/search operations. Test your implementation to verify the predicted complexity bounds are achieved

- Implement the above data structure: please run the main program and check the trail in output text file
- Insert operation: user has the option to insert numbers after the initial read of the input text file is done
- search operation: user has the option to search numbers after initial read of the input text file is done

Verification (reading text input file with different data size):

n	run time (ms)
15	4618
37	8402
96	17654
185	32299