

AEA

Lab 5. Amortized Analysis

March 27, 2026

Amortized Analysis: analysis of a **sequence of operations**.

1. Aggregate analysis: if a sequence of n operations takes $T(n)$ worst-case time in total, the *amortized cost per operation* is $T(n)/n$.

2. Accounting method: assign different charges to each operation. When amortized cost a_i is more than the actual cost c_i of operation i , store the difference $a_i - c_i$ as **credit**. Use credit later to pay for operations s.t. $c_i > a_i$. Credit should never be negative.

3. Potential method: use a potential function.

Homework

1. Consider a dynamic table where you have insertion and deletion. Compute the time complexity of a sequence of n INSERT and DELETE operations (use aggregate, accounting and potential methods).

References:

Cormen et al. Introduction to Algorithms. Section 17.4 Dynamic tables

<https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/AmortizedAnalysis.pdf>

(with demo <https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/DemoDynamicTable.pdf>)

2. Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support SEARCH and INSERT on a set of n elements.

Let $k = \lceil \lg(n + 1) \rceil$, and let the binary representation of n be $\langle n_{k-1}, \dots, n_0 \rangle$. We have k sorted arrays A_0, \dots, A_{k-1} , where the length of array A_i is 2^i . Each array is either full or empty, depending on whether $n_i = 1$ or $n_i = 0$, respectively. The total number of elements held in all k arrays is therefore $\sum_{i=0}^{k-1} n_i 2^i = n$. Although each individual array is sorted, elements in different arrays bear no particular relationship to each other.

a. Describe how to perform the SEARCH operation for this data structure. Analyze its worst-case running time.

b. Describe how to INSERT a new element into this data structure. Analyze its worst-case and amortized running times (two methods).

Deadline: April, 3.