



## DAA - Methodology of Analysis

Advertisements

[Previous Page](#)[Next Page](#)

To measure resource consumption of an algorithm, different strategies are used as discussed in this chapter.

### Asymptotic Analysis

The asymptotic behavior of a function  $f(n)$  refers to the growth of  $f(n)$  as  $n$  gets large.

We typically ignore small values of  $n$ , since we are usually interested in estimating how slow the program will be on large inputs.

A good rule of thumb is that the slower the asymptotic growth rate, the better the algorithm. Though it's not always true.

For example, a linear algorithm  $f(n) = d * n + k$  is always asymptotically better than a quadratic one,  $f(n) = c.n^2 + q$ .

### Solving Recurrence Equations

A recurrence is an equation or inequality that describes a function in terms of its value on smaller inputs. Recurrences are generally used in divide-and-conquer paradigm.




$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \\ aT(\frac{n}{b}) + D(n) + C(n) & \text{otherwise} \end{cases}$$

A recurrence relation can be solved using the following methods –

**Substitution Method** – In this method, we guess a bound and using mathematical induction we prove that our assumption was correct.

**Recursion Tree Method** – In this method, a recurrence tree is formed where each node represents the cost.

**Master's Theorem** – This is another important technique to find the complexity of a recurrence relation.

## Amortized Analysis

Amortized analysis is generally used for certain algorithms where a sequence of similar operations are performed.

Amortized analysis provides a bound on the actual cost of the entire sequence, instead of bounding the cost of sequence of operations separately.

Amortized analysis differs from average-case analysis; probability is not involved in amortized analysis. Amortized analysis guarantees the average performance of each operation in the worst case.

It is not just a tool for analysis, it's a way of thinking about the design, since designing and analysis are closely related.

### Aggregate Method

The aggregate method gives a global view of a problem. In this method, if  $n$  operations takes worst-case time  $T(n)$  in total. Then the amortized cost of each operation is  $T(n)/n$ . Though different operations may take different time, in this method varying cost is neglected.

### Accounting Method

In this method, different charges are assigned to different operations according to their actual cost. If the amortized cost of an operation exceeds its actual cost, the difference is assigned to the object as credit. This credit helps to pay for later operations for which the amortized cost less than actual cost.




This method represents the prepaid work as potential energy, instead of considering prepaid work as credit. This energy can be released to pay for future operations.

If we perform  $n$  operations starting with an initial data structure  $D_0$ . Let us consider,  $c_i$  as the actual cost and  $D_i$  as data structure of  $i^{\text{th}}$  operation. The potential function  $\Phi$  maps to a real number  $\Phi(D_i)$ , the associated potential of  $D_i$ . The amortized cost  $\hat{c}_i$  can be defined by

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Hence, the total amortized cost is

$$\sum_{i=1}^n \hat{c}_i = \sum_{i=1}^n (c_i + \Phi(D_i) - \Phi(D_{i-1})) = c_1 + \Phi(D_n) - \Phi(D_0)$$

## Dynamic Table

If the allocated space for the table is not enough, we must copy the table into larger size table. Similarly, if large number of members are erased from the table, it is a good idea to reallocate the table with a smaller size.

Using amortized analysis, we can show that the amortized cost of insertion and deletion is constant and unused space in a dynamic table never exceeds a constant fraction of the total space.

In the next chapter of this tutorial, we will discuss Asymptotic Notations in brief.

[Previous Page](#)
[Next Page](#)

Advertisements



## Bitdefender Antivirus

50% Off. Buy Now!



Nokia 3.2

وعد هواتف  
نوكيا وعد بجد  
هاتف يتطور دائماً.

اكتشف المزيد

3 سنوات  
تحديثات  
أمنية

سنتين  
نظام التشغيل

[Home](#)

[Jobs](#)

[Coding Ground](#)

[Current Affairs](#)

[UPSC Notes](#)

[Online Tutors](#)

[Whiteboard](#)

[Net Meeting](#)

[Tutorix](#)



Search your favorite tutorials.

[Home](#)

[Jobs](#)

[Coding Ground](#)

[Current Affairs](#)

[UPSC Notes](#)

[Online Tutors](#)

[Whiteboard](#)

[Net Meeting](#)

[Tutorix](#)



Search your favorite tutorials.



[About us](#)

[Terms of use](#)

[Cookies Policy](#)

[FAQ's](#)

[Helping](#)

[Contact](#)

[Home](#)

[Jobs](#)

[Coding Ground](#)

[Current Affairs](#)

[UPSC Notes](#)

[Online Tutors](#)

[Whiteboard](#)

[Net Meeting](#)

[Tutorix](#)



Search your favorite tutorials.