

GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

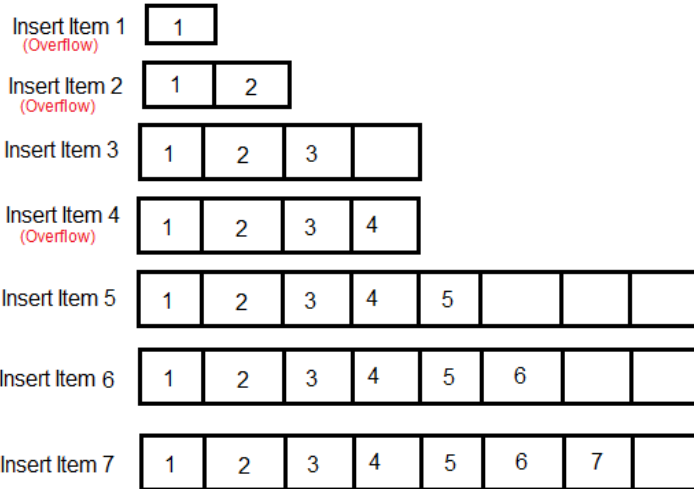
[Graph](#)

Analysis of Algorithm | Set 5 (Amortized Analysis Introduction)

[Amortized Analysis](#) is used for algorithms where an occasional operation is very slow, but most of the other operations are faster. In Amortized Analysis, we analyze a sequence of operations and guarantee a worst case average time which is lower than the worst case time of a particular expensive operation. The example data structures whose operations are analyzed using Amortized Analysis are Hash Tables, Disjoint Sets and Splay Trees.

Let us consider an example of a simple hash table insertions. How do we decide table size? There is a trade-off between space and time, if we make hash-table size big, search time becomes fast, but space required becomes high.

Initially table is empty and size is 0



Next overflow would happen when we insert 9, table size would become 16

The solution to this trade-off problem is to use [Dynamic Table \(or Arrays\)](#). The idea is to increase size of table whenever it becomes full. Following are the steps to follow when table becomes full.

- 1) Allocate memory for a larger table of size, typically twice the old table.
- 2) Cop the contents of old table to new table.
- 3) Free the old table.

If the table has space available, we simply insert new item in available space.

What is the time complexity of n insertions using the above scheme?

If we use simple analysis, the worst case cost of an insertion is $O(n)$. Therefore, worst case cost of n inserts is $n * O(n)$ which is $O(n^2)$. This analysis gives an upper bound, but not a tight upper bound for n insertions as all insertions don't take $\Theta(n)$ time.

Item No.	1	2	3	4	5	6	7	8	9	10
Table Size	1	2	4	4	8	8	8	8	16	16
Cost	1	2	3	1	5	1	1	1	9	1

$$\text{Amortized Cost} = \frac{(1 + 2 + 3 + 5 + 1 + 1 + 9 + 1 \dots)}{n}$$

We can simplify the above series by breaking terms 2, 3, 5, 9.. into two as $(1+1)$, $(1+2)$, $(1+4)$, $(1+8)$

$$\begin{aligned} \text{Amortized Cost} &= \frac{\overbrace{(1 + 1 + 1 + 1 \dots)}^{n \text{ terms}} + \overbrace{(1 + 2 + 4 + \dots)}^{[\log_2(n-1)] + 1 \text{ terms}}}{n} \\ &\leq \frac{[n + 2n]}{n} \\ &\leq 3 \end{aligned}$$

$$\text{Amortized Cost} = O(1)$$

So using Amortized Analysis, we could prove that the Dynamic Table scheme has $O(1)$ insertion time which is a great result used in hashing. Also, the concept of dynamic table is used in [vectors in C++](#).

[ArrayList in Java.](#)

Following are few important notes.

- 1) Amortized cost of a sequence of operations can be seen as expenses of a salaried person. The average monthly expense of the person is less than or equal to the salary, but the person can spend more money in a particular month by buying a car or something. In other months, he or she saves money for the expensive month.
- 2) The above Amortized Analysis done for Dynamic Array example is called **Aggregate Method**. There are two more powerful ways to do Amortized analysis called [Accounting Method](#) and [Potential Method](#). We will be discussing the other two methods in separate posts.
- 3) The amortized analysis doesn't involve probability. There is also another different notion of average case running time where algorithms use randomization to make them faster and expected running time is faster than the worst case running time. These algorithms are analyzed using Randomized Analysis. Examples of these algorithms are Randomized Quick Sort, Quick Select and Hashing. We will soon be covering Randomized analysis in a different post.

Sources:

[Berkeley Lecture 35: Amortized Analysis](#)

[MIT Lecture 13: Amortized Algorithms, Table Doubling, Potential Method](#)

<http://www.cs.cornell.edu/courses/cs3110/2011sp/lectures/lec20-amortized/amortized.htm>

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- [Linearity of Expectation](#)
- [Iterative Tower of Hanoi](#)
- [Count possible ways to construct buildings](#)
- [Build Lowest Number by Removing n digits from a given number](#)
- [Set Cover Problem | Set 1 \(Greedy Approximate Algorithm\)](#)
- [Find number of days between two given dates](#)
- [How to print maximum number of A's using given four keys](#)
- [Write an iterative O\(Log y\) function for pow\(x, y\)](#)



Tweet

3

+1

3

Writing code in comment? Please use ideone.com and share the link here.

5 Comments

GeeksforGeeks

 Login ▾ Recommend Share

Sort by Newest ▾



Join the discussion...

Sriram • 25 days agoHow $1+2+4+8\dots \leq 2n$?? |  • Reply • Share ›**vikky_codder** → **Sriram** • 21 days ago

It forms a G.P. with $A=1$, $R=2$ and $N=\log(n-1)+1$ terms so $\text{sum} = A \cdot (R^N - 1) / (R - 1) = 2^{\log(n-1)+1} - 1 = (2^{\log(n-1)}) \cdot 2 - 1 = (n-1) \cdot 2 - 1 = 2 \cdot n - 3$

 |  • Reply • Share ›**Ashish** → **vikky_codder** • 12 days ago**@vikky_codder** how is it $\lceil \log(n-1) \rceil + 1$ terms ?? |  • Reply • Share ›**vikky_codder** → **Ashish** • 12 days ago

cost is other than one only if item no is of form $(2^{(x-1)} + 1)$ where $x=1, 2, \dots$ so if no of items is n then $(2^{(x-1)} + 1) = n \Rightarrow x = \log(n-1) + 1$

 |  • Reply • Share ›**jsqihui** • 3 months ago

Shouldn't the cop in "2) Cop the contents of old table to new table" be copy?

 |  • Reply • Share › Subscribe Add Disqus to your site Privacy

-
-
-
-

- [Interview Experiences](#)
- [Advanced Data Structures](#)

- [Dynamic Programming](#)
- [Greedy Algorithms](#)
- [Backtracking](#)
- [Pattern Searching](#)
- [Divide & Conquer](#)
- [Mathematical Algorithms](#)
- [Recursion](#)
- [Geometric Algorithms](#)

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

• Recent Comments

- [It_k](#)

i need help for coding this function in java...

[Java Programming Language](#) · [1 hour ago](#)

- [Piyush](#)

What is the purpose of else if (recStack[*i])...

[Detect Cycle in a Directed Graph](#) · [1 hour ago](#)

- [Andy Toh](#)

My compile-time solution, which agrees with the...

[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [1 hour ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [1 hour ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [2 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [2 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team