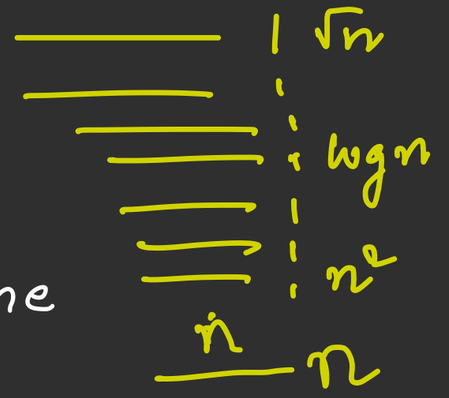
Topics to discuss

1. What is Amortized Analysis
2. Methods for Amortized Analysis.
3. Aggregate Analysis.
4. Example 1: Stack operations.

Amortized Analysis :



In an amortized analysis, we average the time required to perform a sequence of data-structure operations over all the operations performed.

With amortized analysis, we can show that the average cost of an operation is small, if we average over a sequence of operations, even through a single operation within the sequence might be expensive

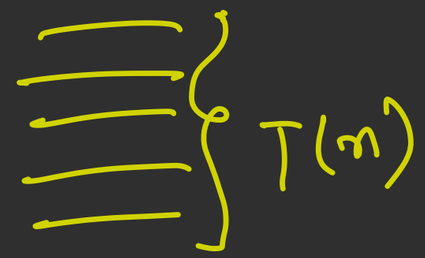
Amortized analysis differs from average-case analysis in that probability is not involved; an amortized analysis guarantees the average performance of each operation in the worst case.

Amortized analysis is used to analyze time complexity of hash table, splay tree, disjoint set etc.

There are three most common techniques used in amortized analysis.

i) Aggregate Analysis / Aggregate Method
ii) Accounting Method
iii) Potential Method.

Aggregate Analysis / Aggregate Method :



In aggregate analysis, we show that for all n, a sequence of n operations takes worst-case time $T(n)$ in total.

In the worst case, the average cost, or amortized cost, per operation is therefore $T(n)/n$.

This amortized cost applies to each operation, even when there are several types of operations in the sequence.

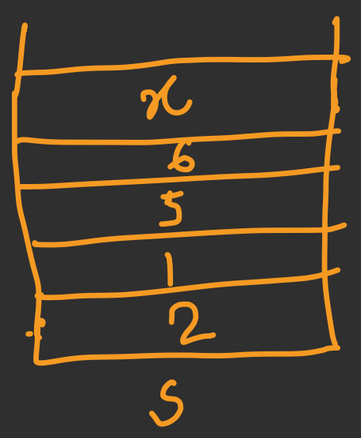
Example 1: Stack Operations :

In our first example of aggregate analysis, we analyze stacks that have been augumented with a new operation.

i) Push (S, x) ⟶ pushes object x onto stack S.

ii) Pop (S) ⟶ pop the top of the stack S and returns the popped object.

iii) Multipop (S, k) ⟶ Remove the k top objects of Stack S.



Pseudocode for Multipop_operation :

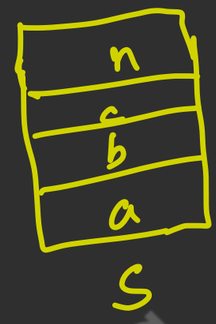
1. while not STACK_EMPTY(S) and K>0
2. POP(S)
3. K = K-1.

Multipop(S, K)

K = 2
K = 1
K = 0

1 push operation ⟶ O(1)

n push operation ⟶ O(n)



Push
Pop
Push
multipop
⋮
Push
multi

1 pop operation ⟶ O(1)

n pop operation ⟶ O(n)

1 multipop operation ⟶ O(n)

n multipop operation ⟶ O(n^2)

Worst case T.C = O(n^2)

Asymptotic Analysis :

Lets analyze a sequence of n Push, POP and Multipop operations on an initially empty stack.

The worst-case cost of a Multipop operation in the sequence is $O(n)$, since the stack size is at most n.

The worst case time of any stack operation is therefore $O(n)$, and hence a sequence of n operations costs $O(n^2)$, since we may have $O(n)$ Multipop operations costing $O(n)$ each.

Although this analysis is correct, the $O(n^2)$ results, which we obtained by considering the worst-case cost of each operation individualy, is not tight.

Amortized Analysis / Aggregate Analysis :

Any sequence of n Push, POP and Multipop operations on an initially empty stack can cost at most O(n).

Explanation : We can pop each object from the stack at most once for each time we have pushed it onto the stack. Therefore, the number of times that POP can be called on a nonempty stack, including calls within Multipop, is at most the number of Push operations, which is at most n.

For any value of n, any sequence of n push, pop and multipop operations takes a total of O(n) time.

The average cost of an operation is $O(n)/n = O(1)$.

In aggregate analysis,
we assign the amortized cost of each operation to be the average cost.
Here, all the three stack operations have an amortized cost of $O(1)$.
Here we are not using any probabilistic reasoning.
we actually showed a worst-case bound of $O(n)$ on a sequence of n operations.
Dividing this total cost by n yielded the average cost per operation, or amortized cost.

Follow Now

Start Practicing

i._am._arfin

Arfin Parween