ASSIGNMENT → 1

HASNAIN ULLAH KHAN
(DSA I)

Q1. What do you mean by amortized analysis? Explain all three suitable method with example.

This analysis is used when the occasional operation is very slow but most of the operations which are executing very frequently are faster. Data structures we need amortized analysis for hash tables.

In a hash table, the most of the time, the searching complexity is $O(1)$. but sometimes it execute $O(n)$ operation. It needs $O(n)$ times operation for collision operation.

The three methods in aggregate analysis are :-

- Potential method
- Aggregate method
- Accounting method.

1. Potential method - According to computational complexity theory, potential method is defined as - "A method implemented to analyze the amortized time & space complexity of a data structure, a measure of its performance over sequence of operatio that eliminate the cost of o infrequent but expensive operation."

The potential technique choose a function ϕ that changes the data structure's states into non negative values, S represent work that has been accounted for in the amortized analysis.

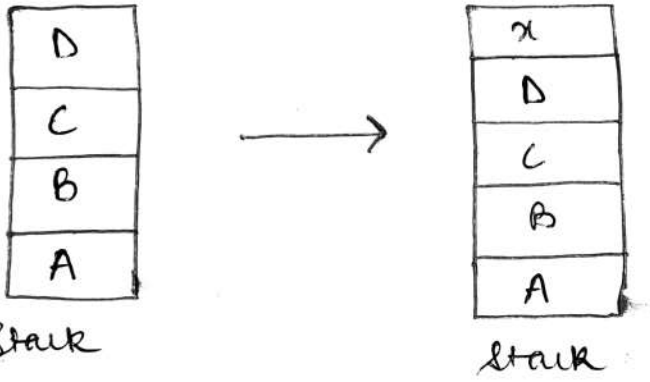
We can design a potential function ϕ on data structure's state if it meets the criteria below:-

1. $\phi(a_0) = 0$, where a_0 is the starting state
2. $\phi(a_t) \geq 0$ for all states of data structure occuring at the time of the course of computation.

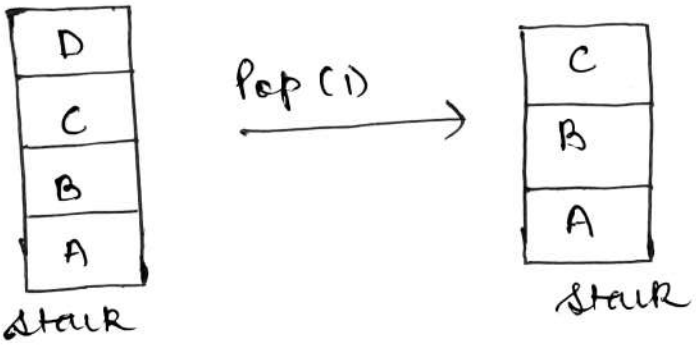
We can design the amortized time of an operation:-
$c + \phi(a') - \phi(a)$, where c is the original cost and a & a' are states of data structure before & after operation.

Analysis of potential method with Example :-
Stack operation :-

Push operation - Time complexity to push an item into a stack is push (x)

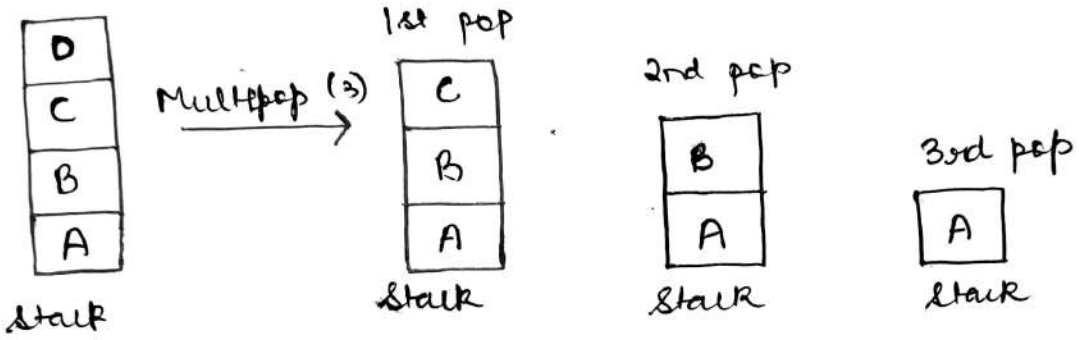


Pop operation - Time complexity to pop an item is O(1)



Multipop operation - Time complexity to pop k item from stack k is min (stack size, k)

Multipop (s, k)
while not stack - empty (s) and k! = 0
do pop (s)
k = k - 1



2. Aggregate method-
This method directly asks on bound on the overall running time of a sequence of operation. In contrast, the accounting method asks to find a payment of number of extra time unit charged to each individual operation.
It involves counting out the complexity of each operation. By expanding each case, one can try to determine a pattern and come up with an overall upper for the algorithm complexity.
- It determines the upper bound $T(n)$ on the total cost of a sequence of n operation then calculate the amortized cost to be $T(n)$.

For a sequence of n operation the cost is:-

$$\frac{\text{cost (n operation)}}{n} = \frac{\text{cost (normal operation)} + \text{cost (exp. operati}}{n}$$

:- Increasing a binary counter

Binary counter of length k, A[0 . . . k-1] of bit array

Increment (A)

1. i <- - 0
2. while i<k and A[i] =1
3. do A[i] <- - 0 (flip, reset)
4. i < i + 1
5. if i<k
6. then A[i] <- - -1 (flip, set)

Initially table is empty and size is 0.

Insert item 1 | 1 |
(overflow)

Insert item 2 | 1 | 2 |
(overflow)

Insert item 3 | 1 | 2 | 3 |
(overflow)

Insert item 4 | 1 | 2 | 3 | 4 |
(overflow)

Insert item 5 | 1 | 2 | 3 | 4 | 5 |
(overflow)

Insert item 6 | 1 | 2 | 3 | 4 | 5 | 6 |
(overflow)

Next overflow would happen when we insert 8 the the table size would become 15.

3. Accounting method-

It is a process in which detailed line item in a financial transaction or statement are carefully examined for a given account, often by a trained auditor or accountant.

The main method adopted for accounting analysis include:-

- Past Performance -
The accounting analysis uses past statistics across historical time period for a single company.

- Future performance -
The accounting analysis is performed by utilising historical figures and certain statistical & mathemati-cal techniques counting present and future value.

- Comparative Performance -
The accounting analysis is also done through comparison btw. similar business companies.

Example :-
An example will help to illustrate the use of the accounting method.

Table expansion -
It is often necessary to create a table before it is known how much speed is needed.

The following pseudo code illustrate the table insertion procedure

```
function table - Insert (T, E)
If num (T) = size (T)
    U := create - table (2 x size (T))
for each f in T
  Elementary - Insert (U, f)
  T := U
  Elementary - Insert (T, E)
```