La "amortized analysis" è una tecnica utilizzata nell'analisi degli algoritmi per calcolare il costo medio complessivo di un'operazione o di un insieme di operazioni in un algoritmo, considerando non solo il costo individuale di ciascuna operazione, ma anche il costo cumulativo nel tempo. L'amortized analysis è particolarmente utile per valutare l'efficienza di algoritmi che coinvolgono strutture dati dinamiche come array dinamici, code, pile e tabelle hash.

Per applicare l'amortized analysis, è comune utilizzare il metodo degli "accounting", che prevede di assegnare un costo virtuale a ciascuna operazione e garantire che il costo complessivo sia sufficiente per coprire tutte le operazioni. Ci sono tre tipi di costi associati a ciascuna operazione:

Costo effettivo (actual cost): Questo è il costo reale dell'operazione. Ad esempio, il costo di un'operazione di inserimento o cancellazione in una struttura dati dinamica.

Costo ammortizzato (amortized cost): Questo è il costo virtuale assegnato a un'operazione in modo che, se sommato a tutti i costi delle operazioni precedenti, copra il costo effettivo dell'operazione e lasci un "credito" eventualmente disponibile per le operazioni future. Il costo ammortizzato può essere distribuito in modi diversi tra le operazioni, ma l'importante è che la somma dei costi ammortizzati sia sempre maggiore o uguale al costo effettivo totale.

Credito residuo (credit or deficit): Questo rappresenta il saldo tra il costo ammortizzato e il costo effettivo. Un credito positivo indica che l'operazione è stata sovvenzionata dalle operazioni precedenti, mentre un credito negativo indica che l'operazione ha contribuito a sovvenzionare le operazioni future.

L'obiettivo nell'uso del metodo degli "accounting" è garantire che il saldo complessivo dei costi ammortizzati e dei crediti residui sia sempre non negativo. Questo significa che il costo totale di eseguire una sequenza di operazioni è almeno tanto quanto il costo effettivo complessivo. In altre parole, l'amortized analysis assicura che l'algoritmo sia efficiente su un lungo periodo.

Questo metodo è particolarmente utile in situazioni in cui alcune operazioni possono richiedere più tempo rispetto ad altre, ma è possibile bilanciare i costi in modo che l'efficienza complessiva dell'algoritmo sia accettabile.

Un esempio comune di applicazione del metodo degli "accounting" è l'analisi delle operazioni in un array dinamico (come quelle in un vettore dinamico o in un ArrayList) o nell'implementazione di code e pile utilizzando array. La distribuzione dei costi ammortizzati tra le operazioni di inserimento, rimozione e accesso assicura che il costo medio sia accettabile, anche se alcune operazioni individuali possono essere costose.

Il "potential method" è un altro approccio utilizzato nell'analisi ammortizzata degli algoritmi. Invece di assegnare costi virtuali direttamente alle operazioni, come nel metodo degli "accounting", il "potential method" utilizza una funzione di "potenziale" o "energia" per valutare il costo ammortizzato.

Ecco come funziona il "potential method" nell'ambito dell'analisi ammortizzata:

1. Si definisce una funzione di potenziale che assegna un valore non negativo a uno stato o a una configurazione dell'algoritmo.
2. Si assegna un "potenziale iniziale" all'inizio dell'esecuzione dell'algoritmo.
3. Per ciascuna operazione eseguita, si calcola il costo effettivo dell'operazione e si sottrae dal potenziale attuale il costo effettivo dell'operazione (cioè, si "spende" parte del potenziale).
4. Il costo ammortizzato di un'operazione è quindi il costo effettivo dell'operazione più la differenza tra il potenziale prima dell'operazione e il potenziale dopo l'operazione.
5. L'obiettivo è garantire che il potenziale totale non diventi mai negativo durante l'intera sequenza di operazioni. In altre parole, il potenziale deve essere sufficiente per coprire i costi effettivi complessivi delle operazioni.

Il "potential method" è particolarmente utile quando è difficile assegnare costi ammortizzati direttamente alle operazioni o quando l'obiettivo è analizzare l'allocazione di risorse in modo più generico.

Un esempio comune in cui il "potential method" può essere applicato è l'analisi di strutture dati come code con operazioni di enqueue e dequeue. Si può utilizzare il potenziale per valutare l'equilibrio tra l'efficienza di enqueue e dequeue in modo da garantire che l'operazione globale rimanga efficiente su un periodo più lungo.

In sintesi, il "potential method" è un'alternativa al metodo degli "accounting" per l'analisi ammortizzata che utilizza una funzione di potenziale per valutare il costo ammortizzato delle operazioni e garantire che il costo totale rimanga sotto controllo.