Learn DAA: From B K Sharma

TCS-503: Design and Analysis of Algorithms

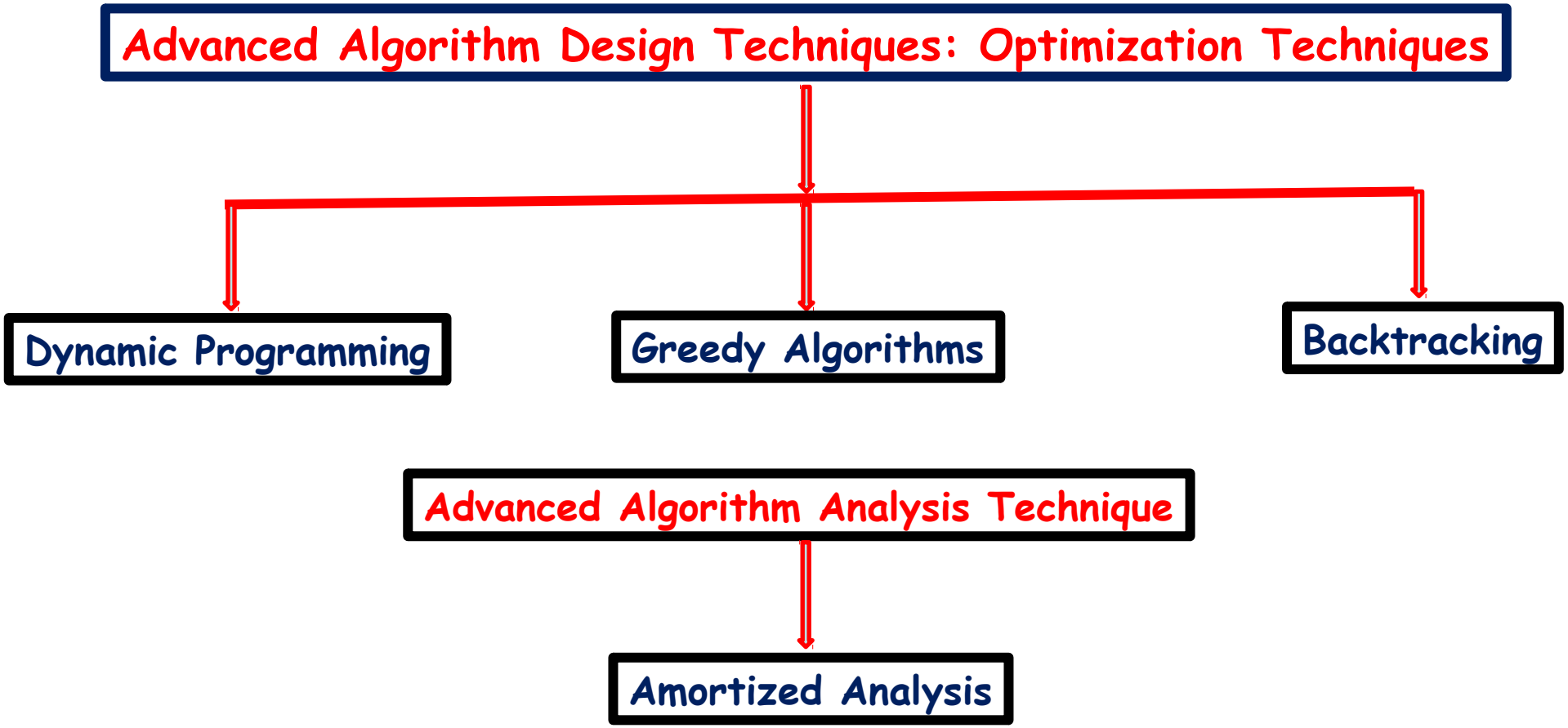
Advanced Design and Analysis Techniques: Dynamic Programming

Learn DAA: From B K Sharma

Unit III

- Advanced Design and Analysis Techniques:
  - Dynamic Programming
  - Greedy Algorithms
  - Amortized Analysis
  - Backtracking.

Learn DAA: From B K Sharma



Learn DAA: From B K Sharma

Amortized Analysis

The term amortization comes from the business world:

Which means

To pay back a debt by making small regular payments over a period of time.

EMI

Equated Monthly Installment

Learn DAA: From B K Sharma

Amortized Analysis

Amortized analysis computes the average time required to perform a sequence of n operations on a data structure.

Difference between Average Case Analysis and Amortized Analysis

To do averages we need to use probability.

For amortized analysis no such assumptions are needed.

In Amortized Analysis, we compute the average cost per operation for any mix of *n* operations.

Learn DAA: From B K Sharma

Amortized Analysis: Three Methods

Aggregate Method

Accounting Method

Potential Method

Amortized Analysis: Three Methods

Aggregate Method

Amortized Cost: $\frac{\text{Total time needed by n-operations, T(n)}}{n}$

Example: Stack Operations

	Actual Cost
PUSH	1
POP	1
MULTIPOP	min(s, k)

T(n)=Total Cost of sequence of n operations=n(1+1+k)=n(2+k)=O(n)

Amortized Cost=T(n)/n=O(n)/n=1=O(1)

Amortized Analysis: Three Methods

Accounting Method

Operations are assigned an amortized cost, associate the extra cost(as a credit with a particular objects [part of] data structure .

Example: Stack Operations

	Actual Cost
PUSH	1
POP	1
MULTIPOP	min(s, k)

	Actual Cost	Amortized Cost
PUSH	1	2
POP	1	0
MULTIPOP	min(s, k)	0

Amortized Analysis: Three Methods

Accounting Method

Total actual Cost of n operations $=n(1+1+k)=n(2+k)=O(n)$

Total Amortized Cost of n Operations$=(2+0+0)=2n=O(n)$

Amortized cost$=T(n)/n=O(n)/n=1=O(1)$

Learn DAA: From B K Sharma

Amortized Analysis: Three Methods

Potential Method

Like the accounting method except that credit is not associated with a particular part (object) of the data structure, but with a pool of potential energy for the entire data structure.

If the i^{th} operation on a stack containing s objects then Potential Change when PUSH, POP and MULTIPOP

operation	actual cost	$\Delta\Phi$	amortized cost
PUSH	1	$(s+1)-s=1$ where s = # of objects initially	$1+1=2$
POP	1	$(s-1)-s=-1$	$1-1=0$
MULTIPOP	$k'=\min(k,s)$	$(s-k')-s=-k'$	$k'-k'=0$

Amortized Analysis: Three Methods

Potential Method

Total Amortized Cost of n operations=n(actual cost + Potential Change)
=n(amortized cost)
=n(2+0+0)=2n=O(n)

So amortized cost of each operation=T(n)/n=O(n)/n=1= $O(1)$

Amortized Analysis: Three Methods

END