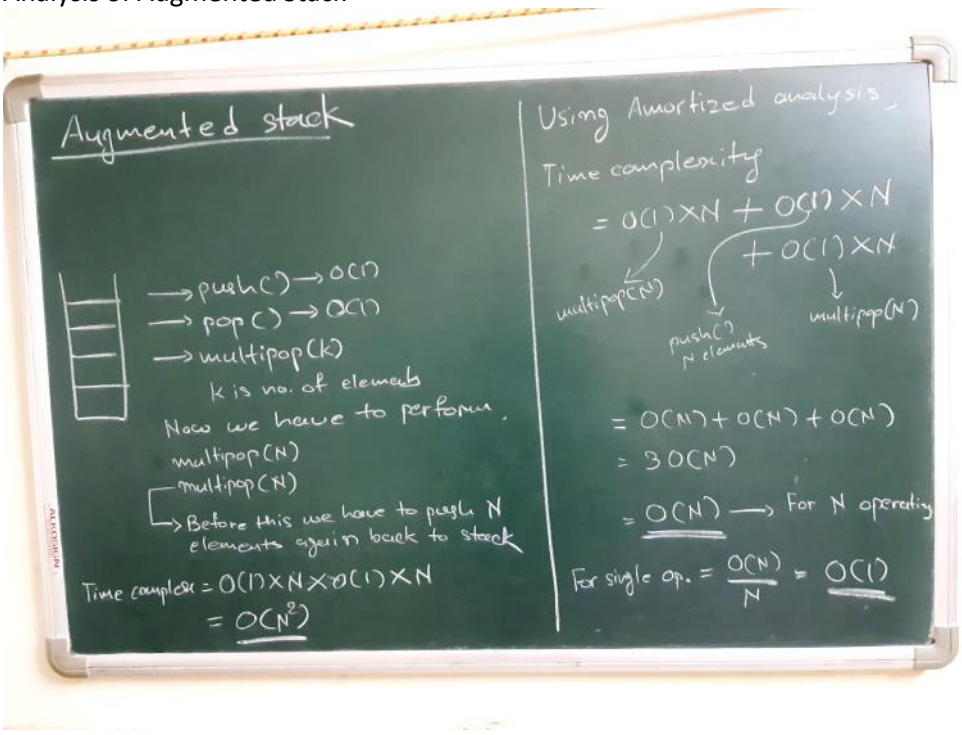
Amortized analysis

Tuesday, December 13, 2022 4:51 PM

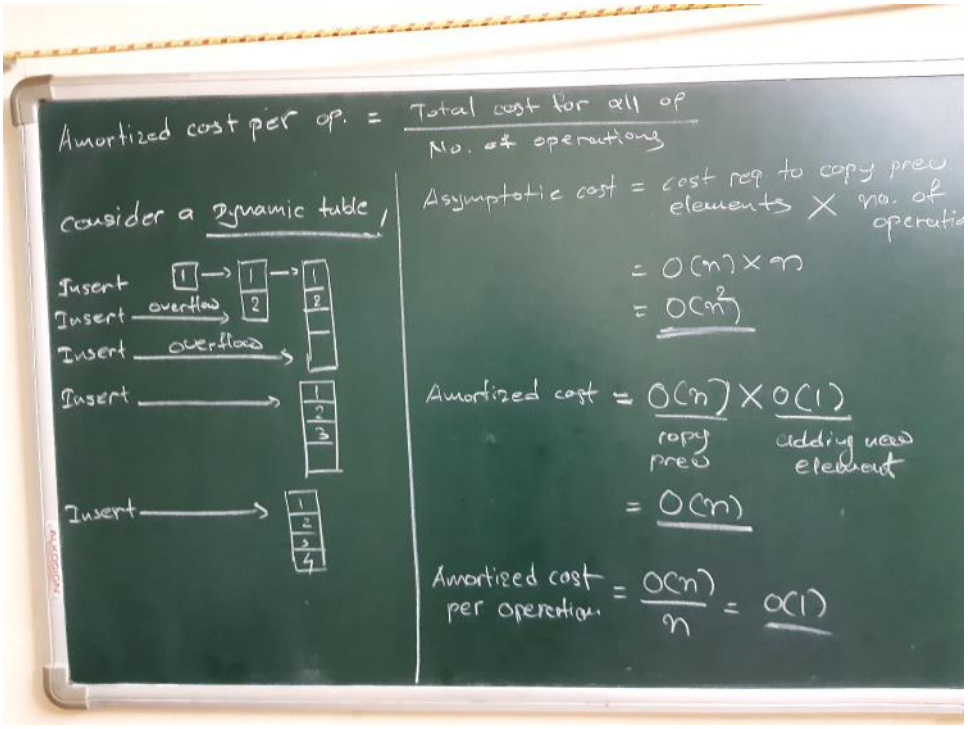
Amortized Analysis -

- If there are some operations happening over a data structure and one operation has very high cost or time complexity than other operations then worst case time complexity using asymptotic analysis will not give accurate solution.
- For the cases where some operation costs very high then Amortized analysis is used to get accurate results.
- Asymptotic analysis multiplies number of operation with the worst case complexity and gives this as final solution
- For example a store has 501 items in which 500 items has cost Rs.1 and 1 item having cost Rs.500
- We wanted to find how much money we should take with us to buy all the items.
- Using asymptotic analysis - Highest cost/worst case = 500 and number of items = 501
  Hence solution is 500*501=250,500
- Using amortized analysis - 1 item with cost 500 and 500 items with cost 1
  Hence solution is 1*500+500*1 = 1,000
- Hence Amortized analysis has given better solution than asymptotic analysis

Analysis of Augmented Stack -

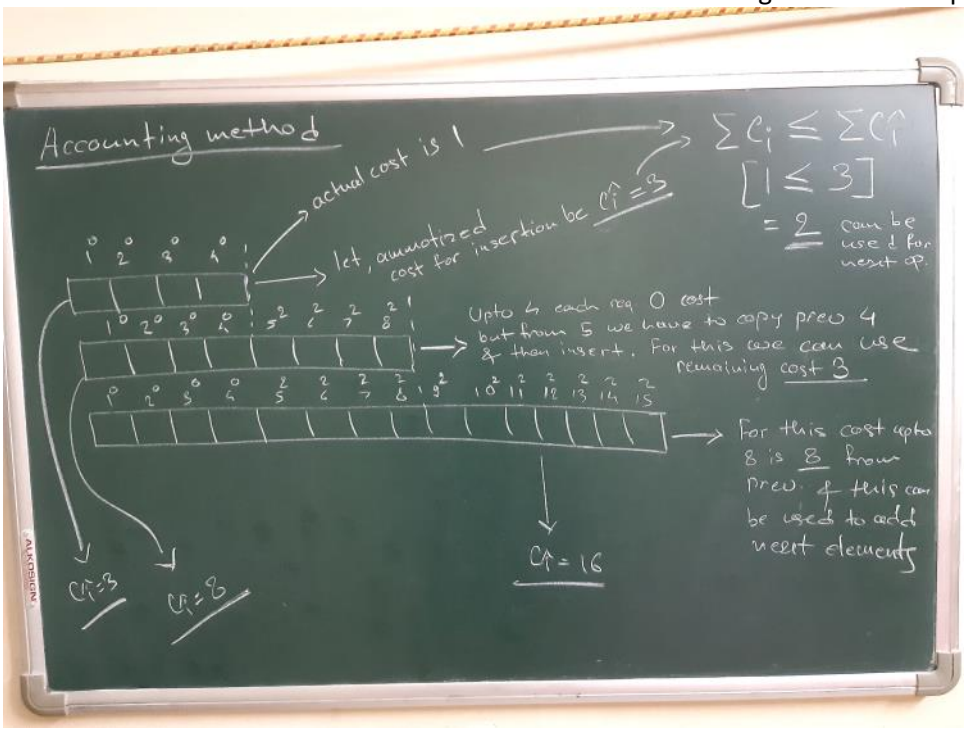


Aggregate method -



Accounting method -

- Assign a random cost as ammotized cost to the 1st operation
- If actual cost is less than ammotized cost then use the remaining cost for next operaitons



Potential method -

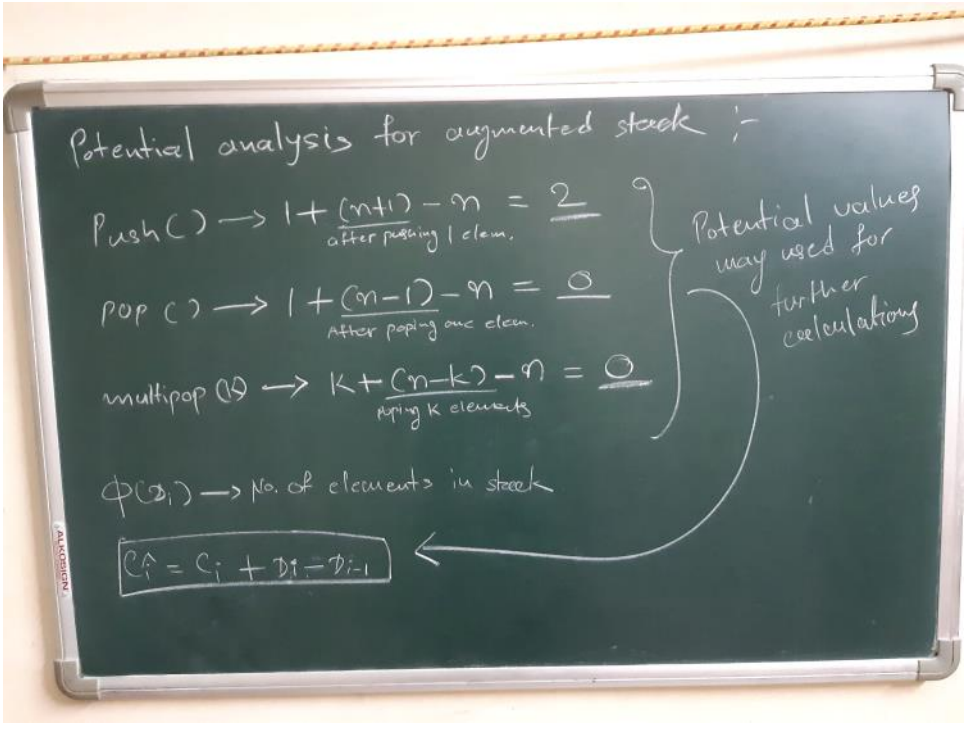
- At any given time there is potential in a data structure
- Initial potential = D0
- Operation 'i' will change potential from D(i-1) to Di
- Actual cost = Ci and Ammotized cost = Ci^
- Potential function are pre-defined function calculated after analysis

- Let it be phi(D0)
- Potential method says that,

  Amotized cost = Actual cost + Change in potential

  Ci^ = Ci + (D(i) - D(i-1))

  Ci^ = Ci + delta(Di)



Tractable problems -

A problem that is solvable by a polynomial-time algorithm.
The upper bound is polynomial.
Here are examples of tractable problems (ones with known polynomial-time algorithms):
– Searching an unordered list
– Searching an ordered list
– Sorting a list
– Multiplication of integers (even though there's a gap)
– Finding a minimum spanning tree in a graph (even though there's a gap)

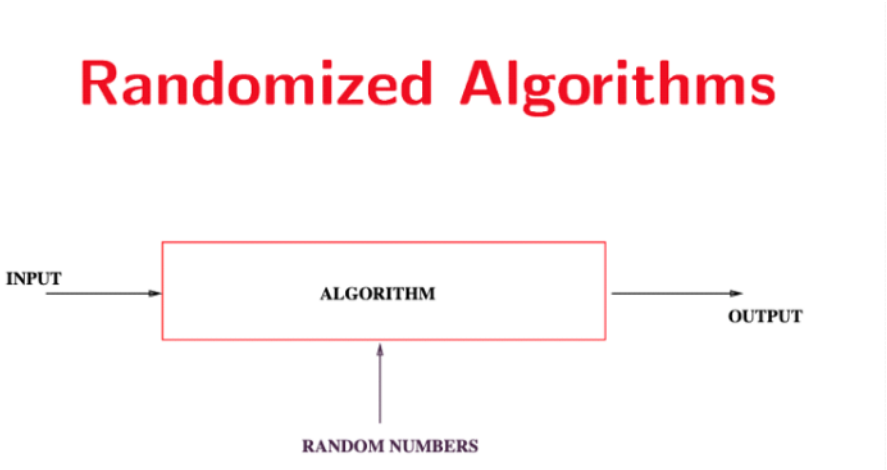
Non tractable problems -

- A problem that cannot be solved by a polynomial-time algorithm.
- The lower bound is exponential.
- From a computational complexity stance, intractable problems are problems for which there exist no efficient algorithms to solve them.
- Most intractable problems have an algorithm that provides a solution, and that algorithm is the brute-force search.
- This algorithm, however, does not provide an efficient solution and is, therefore, not feasible for computation with anything more than the smallest input.
- Towers of Hanoi: we can prove that any algorithm that solves this problem must have a worst-

case running time that is at least 2^n − 1.

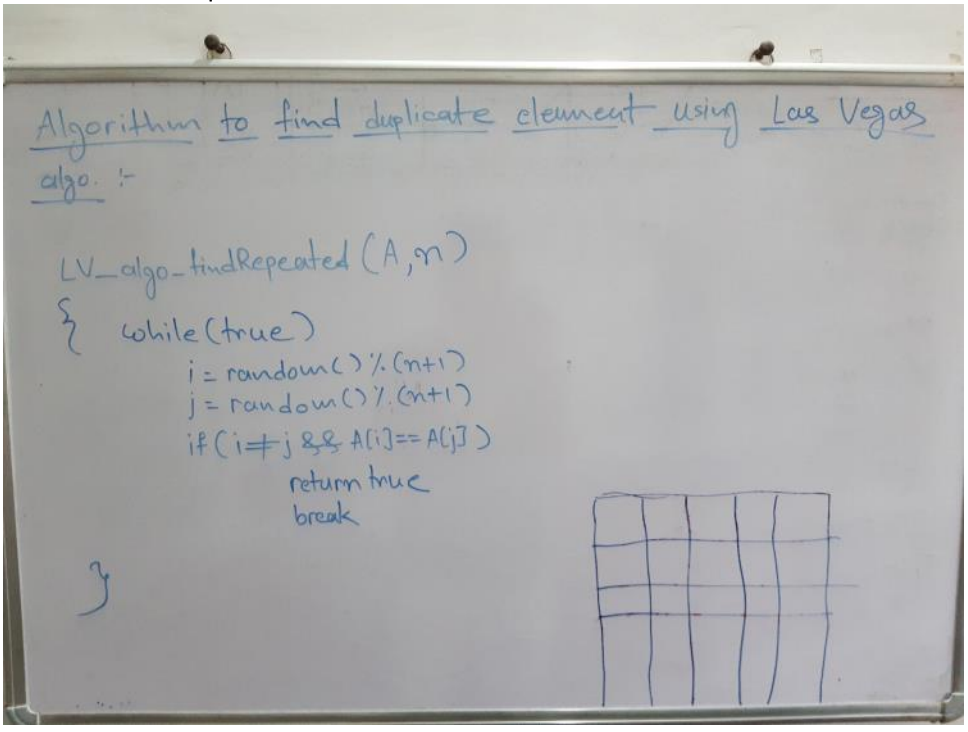
Randomized algorithm -

- An algorithm that uses random number to decide what to do next anywhere in its logic is known as Randomized algorithm
- Used to reduce space and time complexity
- Output may vary even the input is same
- Hence it is a non-deterministic algorithm
- It is also called as probabilistic algorithm



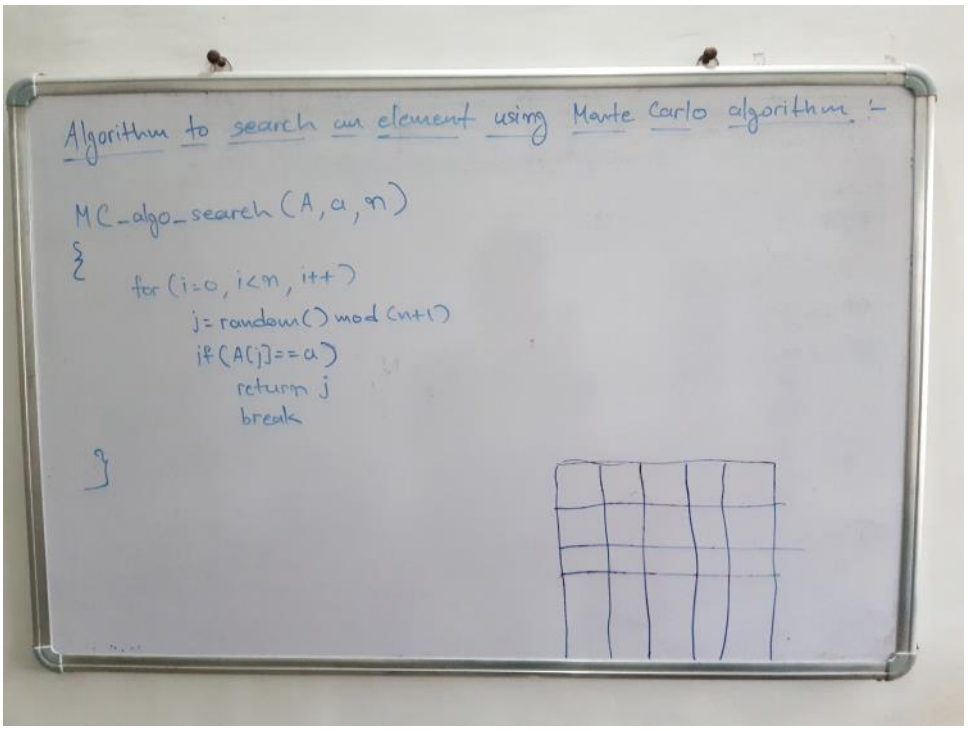
Las Vegas Algorithm -

Output is always correct.
Always gives same output for same input.
Ex. Randomized quick sort

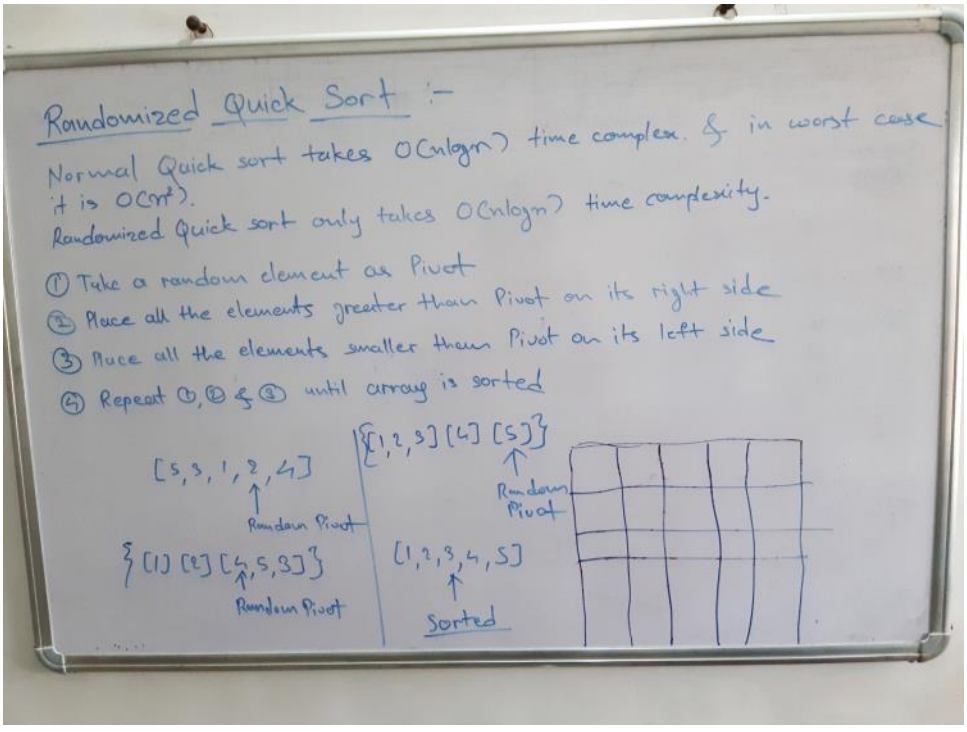


Monte Carlo algorithm -

Output may be incorrect.
May generate different output for same input.
Ex. Randomized median



Randomized Quick Sort -



Approximation algorithm -

- It is the way of dealing with the NP Complete problems for optimization.
- The goal is to generate solution close to the optimal solution.
- It does the same in polynomial time.
- C = Cost of solution
  C* = Cost of optimal solution
  Rho(n) = Approximation ratio

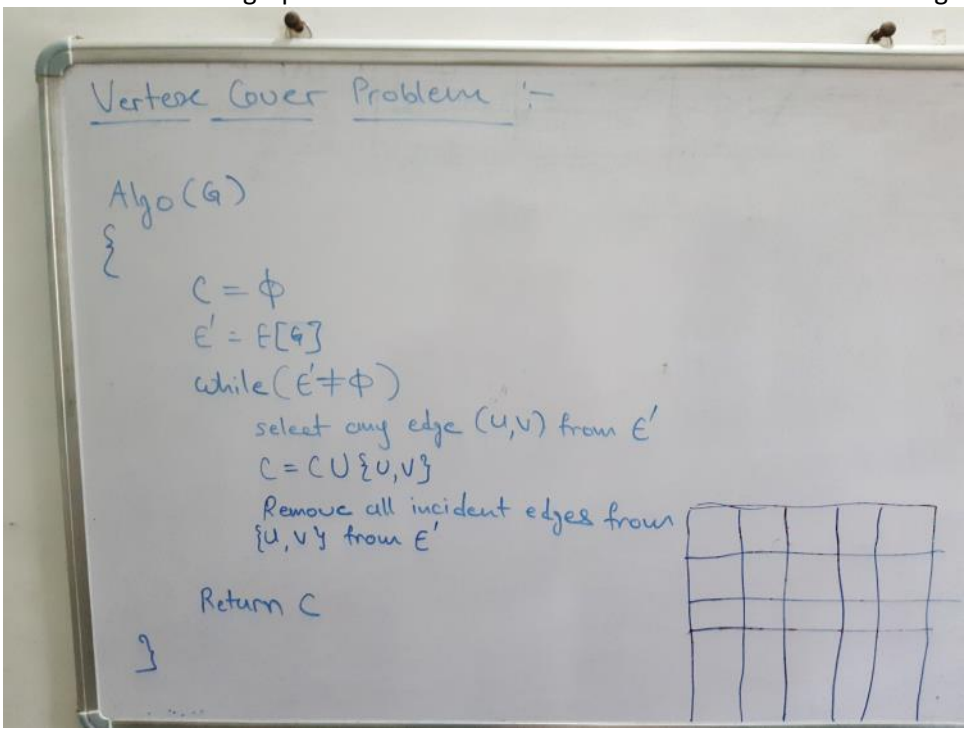
  Maximization - C*/C <= Rho(n)

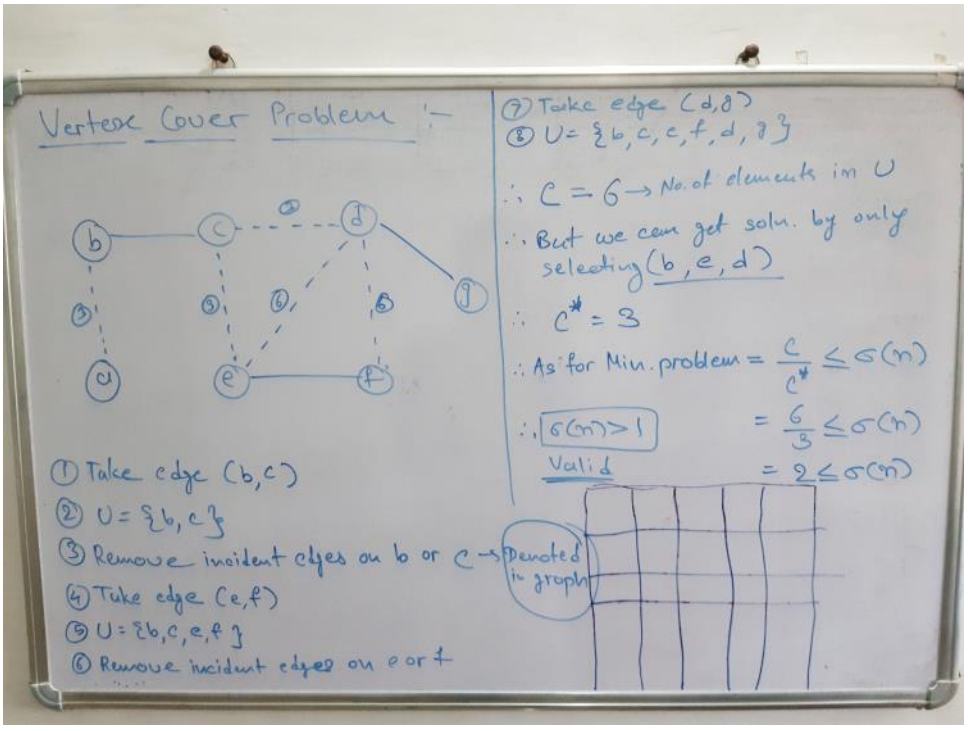
  Minimization - C/C* <= Rho(n)

  Rho(n) > 1

  Vertex cover problem -

  A vertex cover for a graph is the set of minimum vertices that covers all the edges in the graph.





Embedded Systems -

- The computer hardware having software embedded in it is known as Embedded system
- It is a microcontroller or microprocessor based systems
- They are made for specific functionalities
- They are cheapest
- They have low power consumption
- Only one functionality
- They have very strict constraints to work
- Work in real time enviroment to generate resluts

Embedded algorithms -

The algos implemented on a microcontroller or microprocessor based systems are known as Embedded algos
They are similar to normal algos but they have to satisfy the stricter constraints of embedded systems.

Embedded Scheduling Algo -

- Scheduler is the software that decides which operation to do next
- Scheduling algo is the logic or implementation of the Scheduler
- For embedded system a scheduling algo should be microprocessor based and deadline based
- It has to operate in real time to generate the result
- Factors on which scheduling is done
  1.Interdependencies of tasks
  2.Resources
  3.CPU utilization
  4.Deadlines
  5.Cost of the operation

Embedded sorting algo -

It is similar to any normal sorting algo just it has to consider following factors,
1.sort inplace
2.Iterative
3.minimum code size
4.feasible time complexity

Ex. Insertion sort
Time complexity - O(n) for best case, O(n^2) for worst case