Course: Advanced Algorithm and Problem Solving

Week 10 : Advanced Topics in Algorithms

Lemlem Kassa (Ph.D.)

Addis Ababa Science and Technology University, Ethiopia

May, 2025

WEEK 10 : Advanced Topics in Algorithms

Content

- Introduction to approximation algorithms
- Types of Approximation Algorithm
- Randomized algorithms
    - Las Vegas Algorithm
    - Monte Carlo Algorithms
- Online algorithms
- Parallel and distributed algorithms

Lecture Learning Outcome

- Understand approximation algorithms
- Differentiate features of approximation algorithm
- Understand approximation algorithms such as vertex cover problem
- Understand random algorithm
- Understand online algorithm
- Differentiate Parallel and distributed algorithms

Introduction to Approximation Algorithms

- Approximation algorithms are algorithms designed to solve problems that are not solvable in polynomial time for approximate solutions. These problems are known as NP complete problems.
- These problems are significantly effective to solve real world problems, therefore, it becomes important to solve them using a different approach.
- NP complete problems can still be solved in three cases: the input could be so small that the execution time is reduced, some problems can still be classified into problems that can be solved in polynomial time, or use approximation algorithms to find near-optima solutions for the problems.
- This leads to the concept of performance ratios of an approximation problem.

[1]. Vertex Cover Algorithm, Tutorials Point.
https://www.tutorialspoint.com/data_structures_algorithms/vertex_cover_algorithm.htm

Introduction to Approximation Algorithms …cont'd

Features of Approximation Algorithm:

- An approximation algorithm guarantees to run in polynomial time though it does not guarantee the most effective solution.
- An approximation algorithm guarantees to seek out high accuracy and top quality solution(say within 1% of optimum)
- Approximation algorithms are used to get an answer near the (optimal) solution of an optimization problem in polynomial time.
- Because exact solutions are impractical for large inputs.

- Problems that can be solved in polynomial time are considered "tractable" or "easy" to solve, while those that require exponential time are considered "intractable" or "hard".

[2]. Approximation Algorithms, GeeksforGeeks, 2022. https://www.geeksforgeeks.org/approximation-algorithms/

Introduction to Approximation Algorithms …cont'd

Performance Ratios for approximation algorithms :

Scenario-1 :

- Suppose that we are working on an optimization problem in which each potential solution has a cost, and we wish to find a near-optimal solution.
- Depending on the problem, we may define an optimal solution as one with maximum possible cost or one with minimum possible cost, i.e., the problem can either be a maximization or minimization problem.
- We say that an algorithm for a problem has an appropriate ratio of P(n) if, for any input size n, the cost C of the solution produced by the algorithm is within a factor of P(n) of the cost C* of an optimal solution as follows.

$$\max(C/C^*, C^*/C) <= P(n)$$

Introduction to Approximation Algorithms …cont'd

Scenario-2 :

- If an algorithm reaches an approximation ratio of P(n), then we call it a P(n)-approximation algorithm.
- For a maximization problem, 0< C < C*, and the ratio of C*/C gives the factor by which the cost of an optimal solution is larger than the cost of the approximate algorithm.
- For a minimization problem, 0< C* < C, and the ratio of C/C* gives the factor by which the cost of an approximate solution is larger than the cost of an optimal solution.

- Performance ratios (**approximation ratios**) quantify how close an approximation algorithm's solution is to the optimal solution.
- They are crucial for analyzing the quality of approximation algorithms for **NP-hard optimization problems**.

Types of Approximation Algorithm

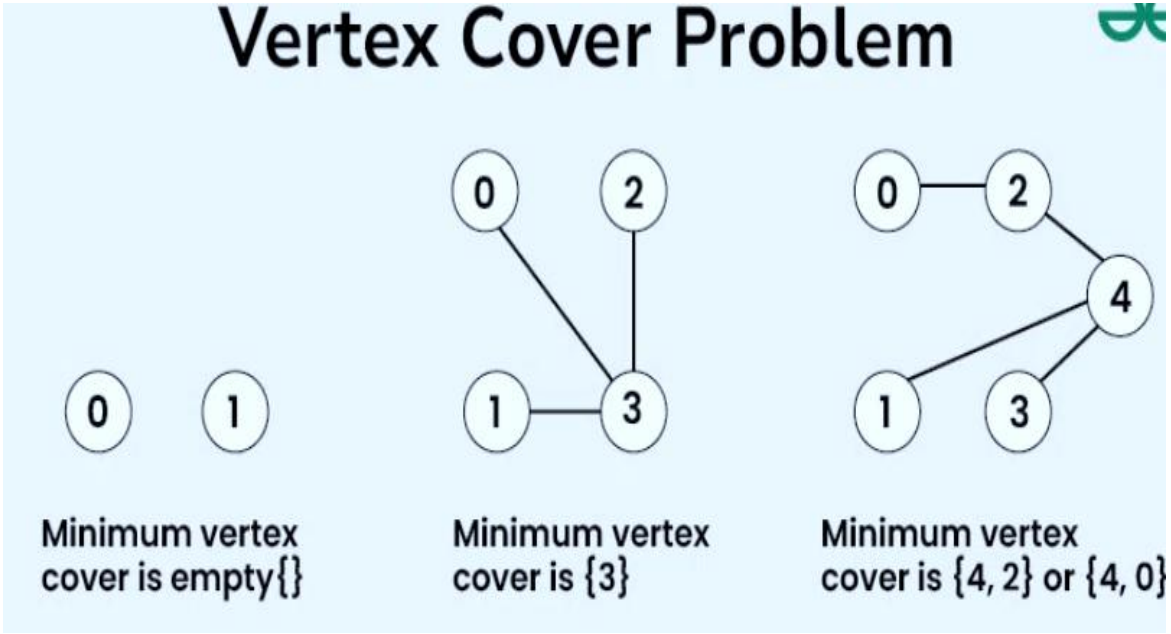
The Vertex Cover Problem

- In the vertex cover problem, the optimization problem is to find the vertex cover with the **fewest vertices**, and the approximation problem is to find the vertex cover with **few vertices**.
- A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either ‘u’ or ‘v’ is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph.
- Given an undirected graph, the vertex cover problem is to find minimum size vertex cover.

[3]. Introduction and Approximate Solution for Vertex Cover Problem, GeeksforGeeks, 2024. https://www.geeksforgeeks.org/introduction-and-approximate-solution-for-vertex-cover-problem/

Types of Approximation algorithm …cont'd

- Examples of Vector cover Problems

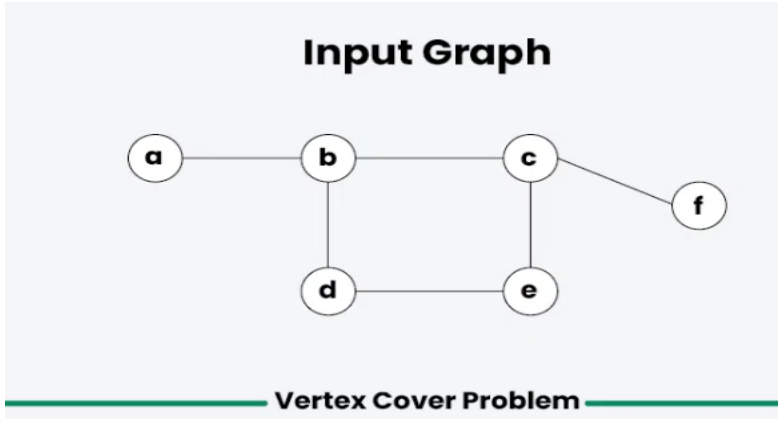


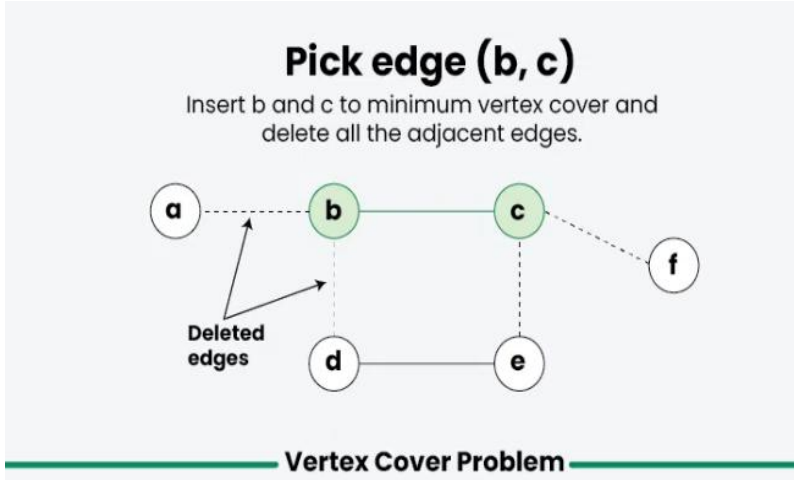
[3]. Introduction and Approximate Solution for Vertex Cover Problem, GeeksforGeeks, 2024. https://www.geeksforgeeks.org/introduction-and-approximate-solution-for-vertex-cover-problem/

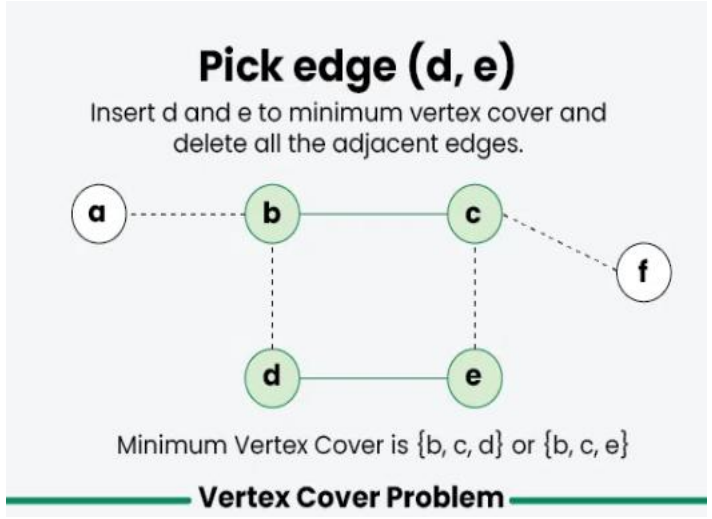
Types of Approximation algorithm …cont'd

- **Approximate Algorithm for Vertex Cover**

```
1) Initialize the result as {}
2) Consider a set of all edges in given graph.  Let
the set be E.
3) Do following while E is not empty
...a) Pick an arbitrary edge (u, v) from set E and add
'u' and 'v' to result
...b) Remove all edges from E which are either
incident on u or v.
4) Return result
```







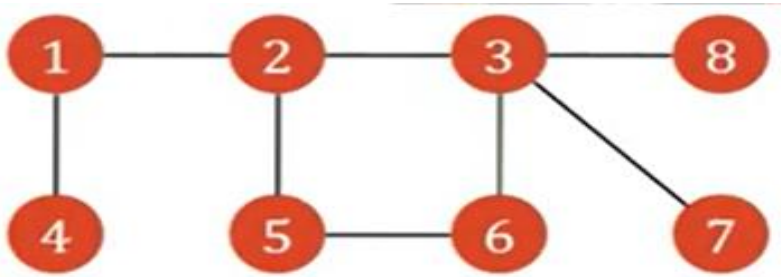
[3]. Introduction and Approximate Solution for Vertex Cover Problem, GeeksforGeeks, 2024. https://www.geeksforgeeks.org/introduction-and-approximate-solution-for-vertex-cover-problem/

Types of Approximation algorithm …cont'd

Consider the following Graph to find Minimum size vertex cover

Example 2

```
APPROX-VERTEX-COVER(G)
1  C = ∅
2  E' = G.E
3  while E' ≠ ∅
4      let (u, v) be an arbitrary edge of E'
5      C = C ∪ {u, v}
6      remove from E' edge (u, v) and every edge incident on either u or v
7  return C
```



E'= {
(1,2), (2,3), (1,4), (2,5),
(3,6), (5,6), (3,7), (3,8)
};

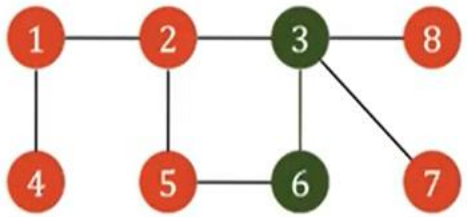
[4]. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009.

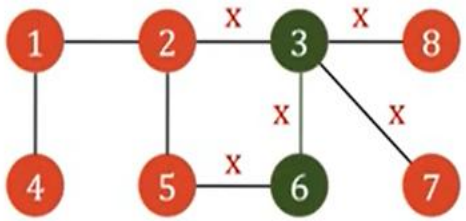
Types of Approximation algorithm …cont'd

Step 1 : choose Arbitrary Edge

$C = \emptyset$

$E' = \{ (1,2), (2,3), (1,4), (2,5), (3,6), (5,6), (3,7), (3,8)\};$





Choose an Arbitrary Edge (3,6) → *Remove all edges associated with vertices 3 & 6*

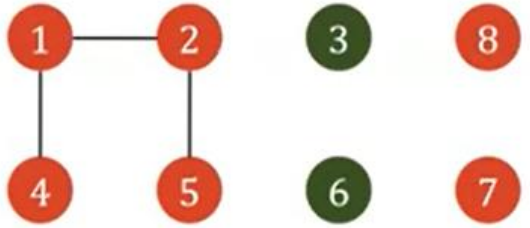
$C \leftarrow C \cup \{(u, v)\} \Rightarrow C \leftarrow \emptyset \cup \{(3, 6)\}$

$\therefore C=\{3, 6\}$

Step 2: Remove all edges associated vertices 3&6

$C = \{3, 6\}$

$E' = \{ (1,2), (1,4), (2,5)\};$



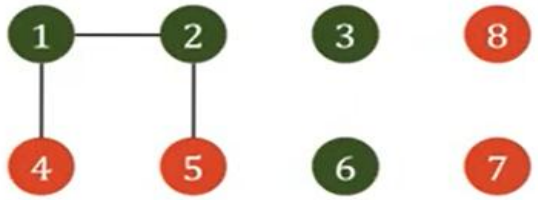
Remove all edges associated with vertices 3 & 6

Types of Approximation algorithm …cont'd

Step 3 : choose another Arbitrary Edge

C = {3, 6, 1, 2}

E'= { (1,2), (1,4), (2,5)};



Choose another Arbitrary Edge (1,2)

$C \leftarrow C \cup \{(u, v)\} \Rightarrow C \leftarrow \{(3, 6)\} \cup \{(1, 2)\}$

$\therefore$ C={3, 6, 1, 2}

Step 4 : Remove all ages associated with Vertices 1 and 2

C = {3, 6, 1, 2}

E'= {Ø};

Choose another Arbitrary Edge (1,2) *Remove all edges associated with vertices 1 & 2*

Therefore, the Minimum size vertex cove C= 3 ,6, 1, 2

Randomized Algorithms

- The world of computer science and mathematics, deterministic algorithms are often preferred due to their predictability and guaranteed correctness. However, there are scenarios where deterministic approaches may be less efficient or impractical.
  - **This is where randomized algorithms come into play.**
- The term “Randomized Algorithm” refers to an algorithm that uses random integers to choose the next step in any part of its logic. So basically it uses random numbers to decide what to do next in its logic.
- Randomized Algorithm uses randomness as a computing tool for algorithm design. That means the program logic is also going to consider randomness as one of the components and we have to use it for finding the solution to the given problem.

- Randomized Algorithms are also called **Probabilistic Algorithms.**
- Example randomize quick sort , we pick random number to pick the next pivot

Randomized Algorithms …cont'd

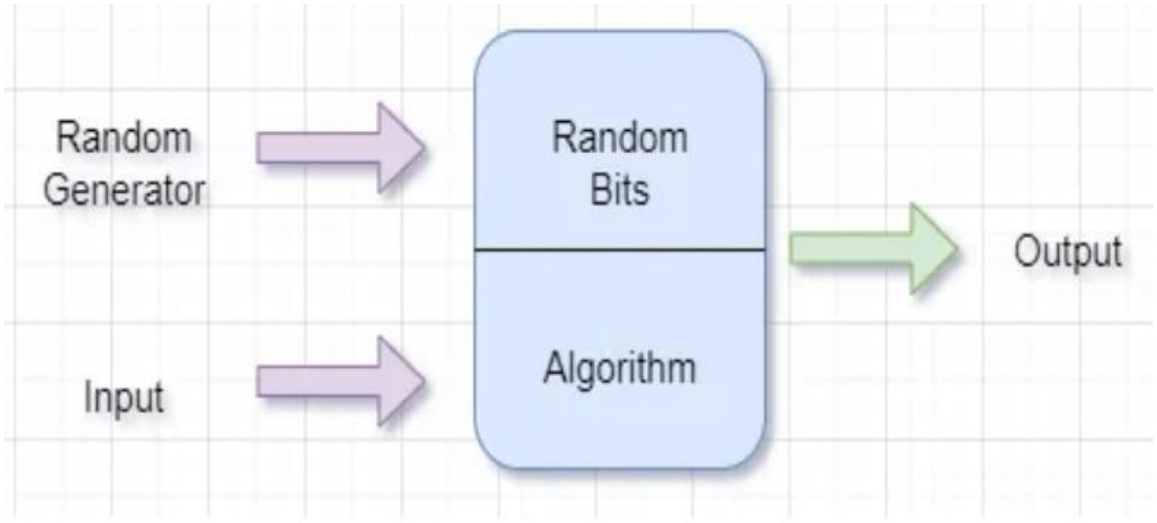
- A Randomized Algorithm makes use of a **Randomizer** (such as a random number generator). So, some of the decisions made in the algorithm depend on the output of the randomizer. The output of any randomizer might differ in an unpredictable way from run to run.
- And this output could also differ from run to run for the same input. Similarly, the execution time could also vary from run to run for the same input.

[5]. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin , Medium, 2023. https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996

Randomized Algorithms …cont'd

Randomized Algorithms depends on two factors:

- **Input** given to the algorithm
- **Random choices** as a part of logic



Basic structure

[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Randomized Algorithms …cont'd

Characteristics of Randomized Algorithms

- The output of the randomized algorithm depends on the random decisions taken by the Randomizer.
- The output is based on Probability. (we can take an example of tossing a coin).
- Neglect the negligible errors that occurred during the long run.

[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Randomized Algorithms …cont'd

Generation of Random Numbers

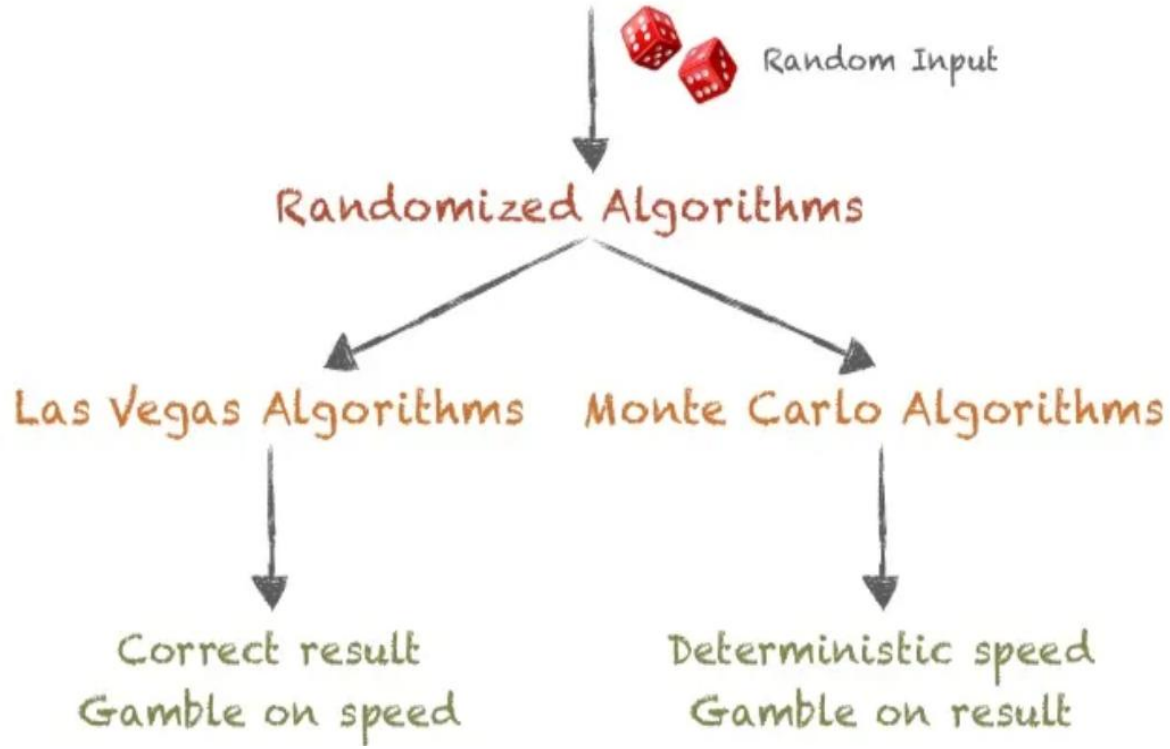
- A random number generator generates a random number.
- There are two types of random numbers:
  - **True random number :** A true random number is generated based on radioactive delay, flipping of coins, etc. Generated number is not recreated again in future. E.g. Simulations, gaming
  - **Pseudo random number :** A pseudo random number is generated using software applications. It can be recreated if the formula is known. It is sufficient for most of the purposes**.** E.g. Simulations, gaming

[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Randomized Algorithms …cont'd

Randomized Algorithms can be categorized into two classes:

- Monte Carlo Algorithms
- Las-Vegas Algorithms



[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Randomized Algorithms …cont'd

Las Vegas Algorithms

- It is randomized algorithms that always provide the correct result, but the running time is subject to randomization.
- In other words, while they guarantee the accuracy of the output, the time it takes to obtain this result may vary***. Quicksort is an example of a Las Vegas algorithm***.

When to Use Las Vegas Algorithms?

a. Approximation Problems: Las Vegas algorithms are ideal when dealing with optimization problems where we need an approximation to a solution. They provide solutions that are typically close to the optimal result while still ensuring correctness.

b. NP-Hard Problems: When dealing with NP-hard problems that are computationally intractable in the worst-case, Las Vegas algorithms can provide practical solutions within a reasonable amount of time.

[7]. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin Basarikatti, Medium, 2023
https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996

Randomized Algorithms …cont'd

Monte Carlo Algorithms

- Monte Carlo algorithms are randomized algorithms that provide a result quickly, but this result may not always be correct.
- They rely on randomization to make probabilistic decisions that might lead to incorrect outputs in some cases.
- However, they can be designed to produce accurate results with high probability.

[7]. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin Basarikatti, Medium, 2023
https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996

Randomized Algorithms …cont'd

When to Use Monte Carlo Algorithms?

a. Problems with Tolerable Error: Monte Carlo algorithms are appropriate for problems where a small probability of error is acceptable. These algorithms are widely used in simulations, statistical analysis, and optimization.

b. High Computational Complexity: When dealing with problems that are extremely time-consuming to solve deterministically, Monte Carlo algorithms can provide relatively quick results.

[7]. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin Basarikatti, Medium, 2023
https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996

Randomized Algorithms …cont'd

Examples of Monte Carlo Algorithms:

i. Monte Carlo Integration: In numerical analysis, Monte Carlo integration uses random sampling to approximate the value of an integral.

- It provides an estimate with an associated level of confidence.

ii. Primality Testing: Algorithms like the Miller-Rabin primality test use randomness to determine whether a number is likely prime or composite.

- While they may occasionally produce incorrect results, the probability of error can be made extremely low by adjusting the number of iterations.

Randomized Algorithms …cont'd

Monte Carlo Algorithms

- In Monte Carlo Algorithms, the algorithm produces mostly or probably correct output. That means, it **may or may not produce correct output**. The output might differ from run to run for the same input.
- Monte Carlo Algorithms are **polynomial time algorithms**, because the running time is completed within polynomial time.

Las Vegas Algorithms

- In Las Vegas Algorithms, the algorithm always **produces the correct output for the same input**. They are **probably fast algorithms**.
- The Execution time of a Las Vegas algorithm depends on the output of the randomizer.

Randomized Algorithms …cont'd

Advantages of Randomized Algorithms

- **Simplicity :** Simple in nature. Uses pseudo random number generator
- **Very Efficient :** These algorithms are efficient as compared to deterministic algorithms. Deterministic Algorithms take a long time whereas Randomized Algorithms take less time to run.
- **Computational Complexity :** Computational Complexity is better than deterministic algorithms. It also helps to reduce both space as well as time complexity. Because it do not need to store all possible outcomes or paths.

[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Randomized Algorithms …cont'd

Disadvantages of Randomized Algorithms

- **Quality :** Randomized Algorithms depends on the quality of random number generator used as part of the algorithms
- **Reliability :** Some algorithms may lack certain guaranteed solutions. We may not firmly say that we will get the correct solution.
- **Hardware Failure :** Sometimes hardware may fail and we may not get the correct solution.

[6]. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93

Online Algorithm

- **Online algorithm** : is one that can process its input piece-by-piece in a serial fashion, i.e., in the order that the input is fed to the algorithm, without having the entire input available from the beginning.
- In contrast, an **offline algorithm** is given the whole problem data from the beginning and is required to output an answer which solves the problem at hand.

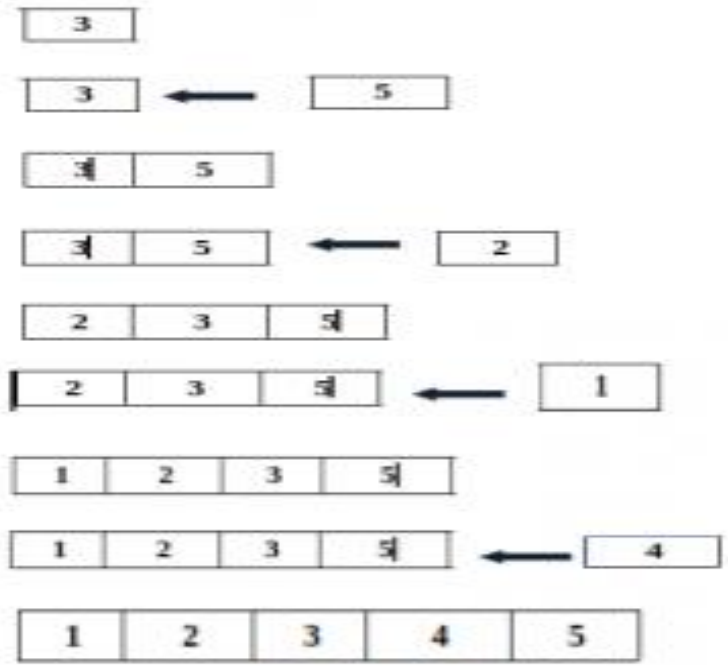
Example:-

- The **selection sort** algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning which requires access to the entire input; it is thus an offline algorithm.

[8]. Online Algorithm, GeeksforGeeks,2017. https://www.geeksforgeeks.org/online-algorithm/

Online Algorithm …cont'd

- On the other hand, insertion sort considers one input element per iteration and produces a partial solution without considering future elements.
- Thus insertion sort is an online algorithm.
- Because an online algorithm does not know the whole input, it might make decisions that later turn out not to be optimal.
- Note that insertion sort produces the optimum result. Therefore, for many problems, online algorithms cannot match the performance of offline algorithms.



[8]. Online Algorithm, GeeksforGeeks,2017. https://www.geeksforgeeks.org/online-algorithm/

Parallel and distributed algorithms

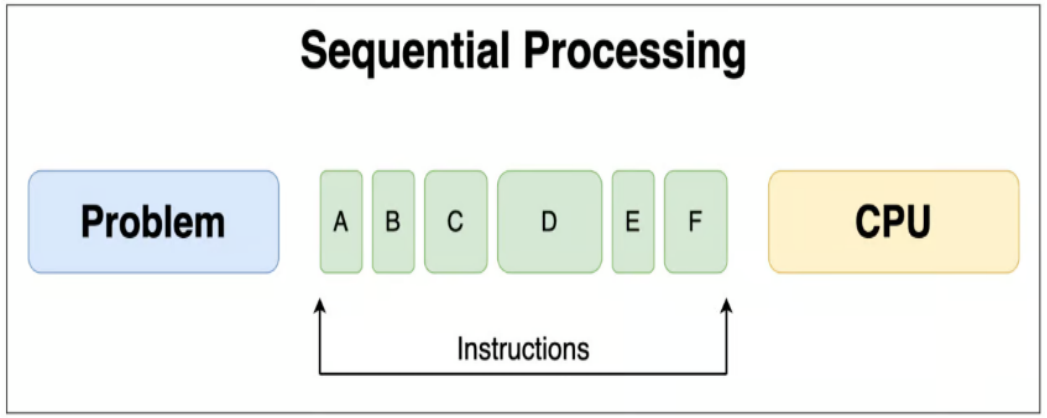
- A **parallel algorithm** is an algorithm that can execute several instructions simultaneously on different processing devices and then combine all the individual outputs to produce the final result.
- **Parallelism** is the process of processing several set of instructions simultaneously. It reduces the total computational time.
- Parallelism can be implemented by using **parallel computers,** i.e. a computer with many processors.
- Parallel computers require parallel algorithm, programming languages, compilers and operating system that support multitasking.

[9]. Sequential vs. Parallel Processing: Which Is Better?,BlockByte, 2024. https://blockbyte.tech/p/sequential-vs-parallel-processing-better

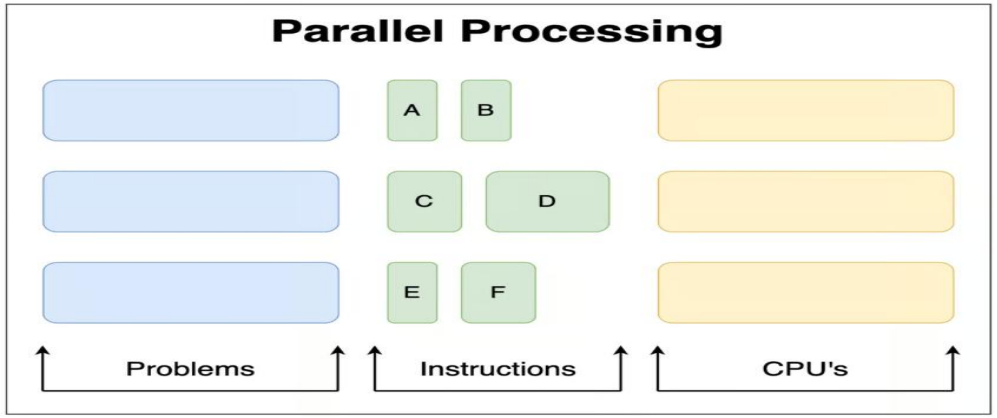
Parallel and distributed algorithms …cont'd

- An **algorithm** is a sequence of instructions followed to solve a problem. While designing an algorithm, we should consider the architecture of computer on which the algorithm will be executed.
- As per the architecture, there are two types of computers:-

Sequential Computer



Parallel Computer



[9]. Sequential vs. Parallel Processing: Which Is Better?,BlockByte, 2024. https://blockbyte.tech/p/sequential-vs-parallel-processing-better

Parallel and distributed algorithms …cont'd

- Depending on the architecture of computers, we have two types of algorithms –
  - **Sequential Algorithm** – An algorithm in which some consecutive steps of instructions are executed in a chronological order to solve a problem.
  - **Parallel Algorithm** – The problem is divided into sub-problems and are executed in parallel to get individual outputs.
  - Later on, these individual outputs are combined together to get the final desired output.

[9]. Sequential vs. Parallel Processing: Which Is Better?,BlockByte, 2024. https://blockbyte.tech/p/sequential-vs-parallel-processing-better

Parallel and distributed algorithms …cont'd

- Selecting a proper designing technique for a parallel algorithm is the most difficult and important task.
- Most of the parallel programming problems may have more than one solution.

Designing techniques for parallel algorithms –

- **Divide and conquer**: In the divide and conquer approach, the problem is divided into several small sub-problems.
- Then the sub-problems are solved recursively and combined to get the solution of the original problem.
- **Greedy Method**: A greedy algorithm works recursively creating a group of objects from the smallest possible component parts. Recursion is a procedure to solve a problem in which the solution to a specific problem is dependent on the solution of the smaller instance of that problem.

Parallel and distributed algorithms …cont'd

Designing techniques for parallel algorithms – …..cont'd

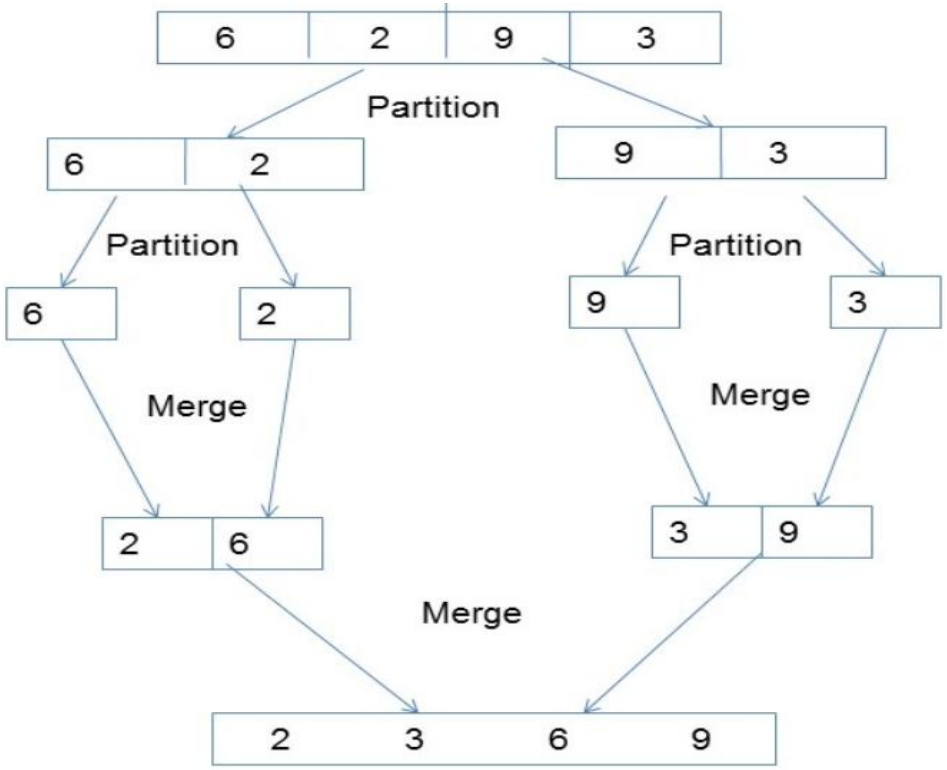
- **Dynamic Programming**: Dynamic programming is an optimization technique, which divides the problem into smaller sub-problems and after solving each sub-problem, dynamic programming combines all the solutions to get ultimate solution.
- **Backtracking**: In backtracking, we start with a possible solution, which satisfies all the required conditions. Then we move to the next level and if that level does not produce a satisfactory solution, we return one level back and start with a new option.

[9]. Sequential vs. Parallel Processing: Which Is Better?,BlockByte, 2024. https://blockbyte.tech/p/sequential-vs-parallel-processing-better

Parallel and distributed algorithms …cont'd

Example : Parallel Algorithm – Sorting

- Merge sort first divides the unsorted list into smallest possible sub-lists, compares it with the adjacent list, and merges it in a sorted order.
- It implements parallelism very nicely by following the divide and conquer algorithm.



Summary

- Randomized algorithms, including Las Vegas and Monte Carlo algorithms, have a crucial role in solving complex problems efficiently when deterministic methods may be impractical or time-consuming.
- By using randomness strategically, they provide solutions that are often both faster and reasonably accurate. However, the trade-off between efficiency and correctness is a key factor in determining when to choose a randomized algorithm.
- Understanding the nature of the problem at hand and the acceptable level of error is essential in making the right choice.
- Understanding when to choose between sequential and parallel processing can significantly impact performance.
- Sequential processing is ideal for simple, linear tasks where predictability and simplicity are essential.
- In contrast, parallel processing excels at handling complex, large-scale problems by leveraging the power of multiple CPU cores.

References

1. Vertex Cover Algorithm, Tutorials Point. https://www.tutorialspoint.com/data_structures_algorithms/vertex_cover_algorithm.htm
2. Approximation Algorithms, GeeksforGeeks, 2022. https://www.geeksforgeeks.org/approximation-algorithms
3. Introduction and Approximate Solution for Vertex Cover Problem, GeeksforGeeks, 2024. https://www.geeksforgeeks.org/introduction-and-approximate-solution-for-vertex-cover-problem/
4. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009.
5. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin , Medium, 2023. https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996
6. Randomized Algorithms. Random numbers should not be generated, Amaan Khatib, Medium, 2022. https://amaankhatib232.medium.com/randomized-algorithms-4e4a7f9b1f93.
7. When to Choose Randomized Algorithms: Understanding Las Vegas and Monte Carlo Algorithms, Shreya Sachin Basarikatti, Medium, 2023 https://medium.com/@sachin.shreya21/when-to-choose-randomized-algorithms-understanding-las-vegas-and-monte-carlo-algorithms-9324d5e9f996
8. Online Algorithm, GeeksforGeeks,2017. https://www.geeksforgeeks.org/online-algorithm/
9. Sequential vs. Parallel Processing: Which Is Better?,BlockByte, 2024. https://blockbyte.tech/p/sequential-vs-parallel-processing-better

Thank You!

For your attention