Standard Complexity Classes Time and space trade-offs in algorithms analysis, recursive algorithms, Algorithmic Strategies

Instructor: Kayode Oladapo
Email: oladapoka@mcu.edu.ng
Education: Ph.D in Computer Science
Website: https://sites.google.com/view/kayodeabiodunoladapo

KNOWLEDGE
INTEGRITY
SERVICE
McPHERSON
UNIVERSITY

Intended Learning Outcomes

- Understand the standard Complexity Classes
- Understand time and space trade-offs in algorithms analysis
- understand recursive algorithms
- Understand algorithmic strategies

Standard Complexity Classes

- Constant Time – O(1)
- Linear Time – O(n)
- Logarithmic Time – O(log n)
- Quadratic Time – O(n^2)
- Polynomial Time – O(n^2), O(n^3)…
- Exponential Time – O(2^n)

Time

TRADE OFF

Space

For example, a hash table provides fast access to data but can consume a lot of memory.

Time and space trade-offs involve making decisions to optimize either time or space complexity. Sometimes, reducing time complexity can increase space complexity, and vice versa. It's a common challenge in algorithm design.

For Example: Memoization in Dynamic Programming can trade space for time.

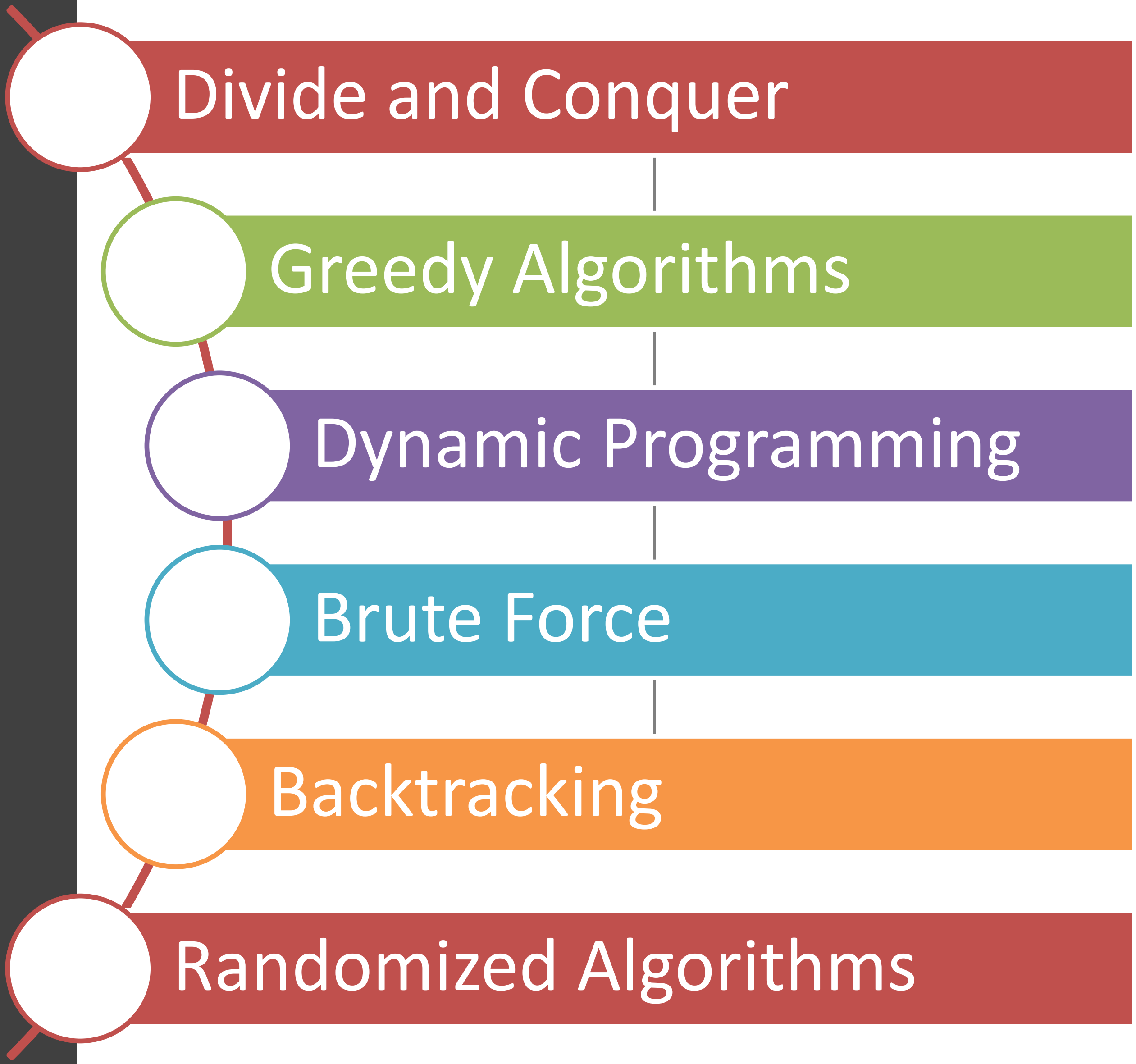
Recursive algorithms are algorithms that solve a problem by solving smaller instances of the same problem.

Example: Merge Sort

Example: Factorial

Recursive is like tree traversal or problems

Algorithmic Strategies



- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Brute Force
- Backtracking
- Randomized Algorithms

Conclusion

Understanding and choosing the right algorithmic strategy is crucial for designing efficient algorithms. Each strategy has its strengths and weaknesses depending on the nature of the problem at hand. Time and space trade-offs are often involved in making these design decisions.

Thank You, See you next week