# DTS203TC
# Design and Analysis of Algorithms

## Revision and Final Exam Information

Dr. Qi Chen
School of AI and Advanced Computing

# Course Contents

- Efficiency of Algorithms and Complexity Measures
- Algorithms and Data Structures
- Computational Intractability and NP-Completeness

# Efficiency of Algorithms and Complexity Measures

- Examples of algorithmic problems and introduction of complexity in terms of various resources
- Asymptotic complexity and notation in conjunction with a discussion on the worst-case versus the average-case complexity

# Efficiency of Algorithms and Complexity Measures

- O-notation
- Ω-notation
- Θ-notation

- Exercise:
  - $n^3 + 3n^2 + 3$ — $O(n^3)$
  - $4n^2 \log n + n^3 + 5n^2 + n$ — $O(n^3)$
  - $2n^2 + n^2 \log n$ — $O(n^2 \log n)$
  - $6n^2 + 2^n$ — $O(2^n)$

# Efficiency of Algorithms and Complexity Measures

- Solving recurrence
  - Substitution method: Guess a bound + Mathematic induction
  - Recursion-tree method: covert the recurrence into a tree
  - Master method

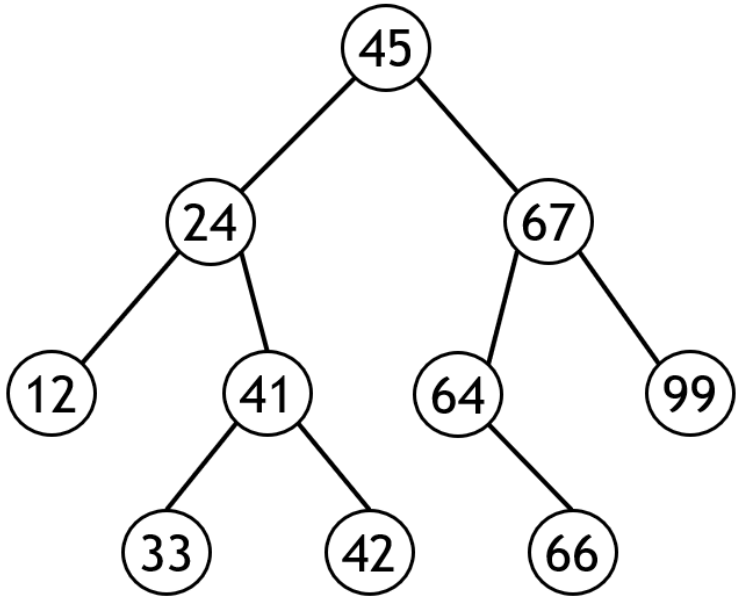
- Exercise:
  - $T(n) = T(n/2) + \Theta(1)$ $\Theta(logn)$
  - $T(n) = 2T(n/2) + \Theta(1)$ $\Theta(n)$
  - $T(n) = 2T(n/2) + \Theta(n)$ $\Theta(nlogn)$
  - $T(n) = 8T(n/2) + \Theta(n^2)$ $\Theta(n^3)$
  - $T(n) = 7T(n/2) + \Theta(n^2)$ $\Theta(n^{log7})$

# Algorithms and Data Structures

- Introduction and analysis of basic data structures with their efficient implementation:
  - Stack
  - Queue
  - LinkedList
  - Hash table
  - Heap

# Algorithms and Data Structures

- Trees:
  - Binary Search Trees
  - Insertion and Deletion
  - Tree Traversal
  - AVL Tree
  - Red-Black Tree




Binary Search Trees

# Algorithms and Data Structures

- Divide and Conquer
  - Merge sort
  - Binary search
- Dynamic Programming
  - Fibonacci numbers
  - Assembly line scheduling problem
  - 0-1 Knapsack problem
  - Longest common subsequence problem
- Greedy Algorithms
  - Activity selection problem
  - Huffman coding problem

# Algorithms and Data Structures

- Graph
  - Graph representation
  - Breadth first search
  - Depth first search
- Minimum Spanning Tree
  - Kruskal's algorithm
  - Prim's algorithm
- Shortest Paths
  - Dijkstra's algorithm
  - Bellman-Ford algorithm
- Maximum Flow
  - Ford-Fulkerson Algorithm
  - Bipartite matching

# Algorithms and Data Structures

- String Matching
  - Naïve algorithm
  - Rabin-Karp Algorithm
  - Knuth-Morris-Pratt Algorithm

# Algorithms and Data Structures

- Number Theory
  - Primes
  - Modular arithmetic
  - Euclid's GCD algorithm
  - Extended Euclid's algorithm
  - Modular Multiplicative Inverse
  - Euler's totient function
  - Powers of an element
  - RSA public-key cryptography

# NP-completeness

- The classes P and NP
- Polynomial-time reduction
- Cook-Levin Theorem
- NP Hard and NP Completeness
- NP completeness problems
  - Hamiltonian circuit
  - SAT (satisfiability)
  - 0/1 knapsack
  - 3-Coloring
  - K-Clique
  - Vertex cover
- How to prove NP Completeness

# Exam

- next Friday (April 12) at M1018 from 14:00 - 16:00.

# Exam

- A closed-book examination.
- Total marks available are 100. (60% of final module mark)
- Only the university approved calculator – Casio FS82ES/83ES – can be used.
- The exam consists of SIX questions.
- NO MCQ
- Only English solutions are accepted.