IT580 Advanced Algorithms

- **Course placement:** Advanced Algorithms is a core course for first year students of M.Tech ICT program.
- **Course format:** It is 3 hours lecture every week.
- **Course content:** This is an algorithms course with an emphasis on teaching techniques for the design and analysis of efficient algorithms, emphasizing methods of application. Topics include divide-and-conquer, randomization, dynamic programming, greedy algorithms, approximation algorithms, complexity theory etc.
- **Assessment method:** one mid-semester examination and a final examination.
- **Course Outcomes:** Students who complete the course should have the ability to do the following:
  - Argue the correctness of algorithms using inductive proofs and invariants.
  - Analyze worst-case running times of algorithms using asymptotic analysis. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms and analyze them.
  - Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Synthesize dynamic-programming algorithms, and analyze them.
  - Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Synthesize greedy algorithms, and analyze them.
– Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.

– Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.

– Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.

– Explain what an approximation algorithm is, and the benefit of using approximation algorithms. Be familiar with some approximation algorithms, including algorithms that are PTAS or FP-TAS. Analyze the approximation factor of an algorithm.

• Topics covered:
  – Basics of algorithm analysis
  – Greedy algorithms
  – Divide and conquer
  – Dynamic programming
  – Graph algorithms
  – NP-completeness
  – Approximation algorithms
  – Randomized algorithms
  – Algebraic algorithms