CS3AC3: Algorithms and Complexity
Graduate Attributes and Indicators

George Karakostas

1 What the students should know and be able to do

1. Students should know and understand
   (a) Analysis of algorithms
   (b) Divide-and-Conquer algorithms
   (c) Greedy algorithms
   (d) Dynamic Programming
   (e) Network Flows
   (f) Turing-Church Thesis, complexity classes (P, NP, PSPACE)
   (g) Simple approximation algorithms
   (h) Simple randomized algorithms
2. Students should be able to
   (a) Analyze the time complexity of algorithms using recurrences
   (b) Design Divide-and-Conquer algorithms
   (c) Design Greedy algorithms
   (d) Design Dynamic Programming algorithms
   (e) Solve Network Flow problems
   (f) Distinguish problems according to their complexity class
   (g) Design simple approximation algorithms
   (h) Use randomness in algorithms

2 Mapping to Attributes with their Indicators

A01 Knowledge	
Competence in Engineering Fundamentals	2a–2h
Competence in specialized engineering knowledge	1a–1h
A02 Analysis	
Ability to identify the essential characteristics of a technical problem, including scope	2a–2h
Ability to identify reasonable assumptions (including identification of uncertainties and imprecise information) that could or should be made before a solution path is proposed	2a–2h
Ability to identify a range of suitable engineering fundamentals (including techniques) that would be potentially useful for analyzing a technical problem	1a–1h, 2a–2h
Ability to decompose and organize a problem into manageable subproblems	2a–2h
Ability to obtain substantiated conclusions as a results of a problem solution, including recognizing the limitations of the solutions	2a–2h
A03 Investigation	
Able to recognize and discuss applicable theory knowledge base	1a–1h,2a–2h
Capable of selecting appropriate model and methods and identify assumptions and constraints	2a–2h
A04 Design	
Recognizes and follows an engineering design process	2a–2h
Recognizes and follows engineering design principles	2a–2h
Properly documents and communicates processes and outcomes	2a–2h
A07 Communication	
Demonstrates an ability to respond to technical and non-technical instructions and questions	1a–1h,2a–2h
Demonstrates appropriate use of technical vocabulary	1a–1h,2a–2h
Constructs effective written arguments	1a–1h,2a–2h

3 Rubrics

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Analysis of algorithms 1a,2a	doesn't understand most concepts %	can perform only simple time analysis %	can form and analyze most time recurrences %	can form and analyze even sophisticated time recurrences %
Divide-and-Conquer & Greedy algorithms 1b,2b,1c,2c	doesn't understand DC & Greedy algorithms %	can analyze but cannot design DC & Greedy algorithms %	can analyze and design DC & Greedy algorithms %	is comfortable with the analysis and design of DC & Greedy algorithms %
Dynamic Programming algorithms 1d,2d	doesn't understand DP algorithms %	can analyze but cannot design DP algorithms %	can analyze and design DP algorithms %	is comfortable with the analysis and design of DP algorithms %
Network Flows 1e,2e	doesn't understand most concepts %	understands basic flow concepts but not flow algorithms %	understands standard flow algorithms %	understands flows and can reduce other problems to flow problems %
Complexity classes and reductions 1f,1f	doesn't understand most concepts %	understands the differences between complexity classes but not reductions %	understands the differences between complexity classes and can mostly do reductions %	understands the differences between complexity classes and can do most reductions %
Approximation algorithms 1g,2g	doesn't understand approximation in algorithms %	understands approximation but cannot analyze approximation algorithms %	understands and can analyze simple approximation algorithms %	has a sophisticated understanding of the design and analysis of approximation algorithms %
Randomized algorithms 1h,2h	doesn't understand randomization in algorithms %	understands probabilities but cannot analyze randomized algorithms %	understands and can analyze simple randomized algorithms %	has a sophisticated understanding of the use of randomization in algorithms %