Soran University
Algorithm Design and Analysis Module Specification

1. Module Title – Algorithm Design and Analysis

2. Module Code - CS306SAD

3. Module Level – Third Stage

4. Module Leader – Thenraja Vettivelraj

5. Teaching Semester – 2nd Semester

6. Credit Rating for the module –4 Credits

7. Prerequisites and co-requisites

Data structures and Algorithms

None

8. Module Summary
[A brief summary of the content of the module, and its assessment, when it takes place and any prerequisites]

This course covers the general design principles of algorithms. It also covers the main techniques of algorithm design and analysis with algorithmic solutions to problems in many domains.

The assessment comprises of three class exam which is of 40% and the final exam will be of 60%. The class exam will be in the 4th week and 8th week of the semester.

9. Module Aims
[A list of the general things that the module aims to achieve in terms of content and what the student will be able to do upon completion of the module]

- Learning different algorithm techniques, design and its analysis.
- Knowledge on how to use a suitable analysis method for any given algorithm.
- Explore on how to design new algorithms for variations of problems.
- Learn different searching techniques and graph algorithms.
- Define formally an algorithmic problem.
- Learn Np-complete problems, Topological problems and Geometric algorithms.

10. Learning Outcomes

[A numbered list of the objectives of the module LO1, LO2, etc. Each outcome should be a specific statement of what the student should be able to do upon completion of the module. Learning objectives should aim to be SMART – Specific, Measurable, Achievable, Realistic]

1) Learning different algorithm techniques, design, and its analysis.
2) Knowledge on how to use a suitable analysis method for any given algorithm.
3) Explore on how to design new algorithms for variations of problems.
4) Learn different searching techniques and graph algorithms.
5) Define formally an algorithmic problem.
6) Learn Np-complete problems, Topological problems and Geometric algorithms.

11. Syllabus

1) **INTRODUCTION [1 session]**
    a. Review of basic data structures (Queue, Stack, link lists, trees,…)
    b. Review of basic algorithms
    c. Recurrences.

2) **Design techniques [3 session]**
    a. Divide-and-Conquer
        i. Multiplication
        ii. Recurrence relations
        iii. Mergesort
        iv. Medians
        v. Matrix multiplication

    b. Prune-and-Search
    c. Dynamic Programming
        i. Shortest paths
        ii. negative cycles
        iii. matrix chain multiplications
        iv. sequence alignment
        v. RNA secondary structure

    d. Greedy Algorithms
        i. Minimum spanning tree
        ii. Union-Find algorithms

iii. Kruskal's Algorithm
iv. Clustering
v. Huffman Codes
vi. Multiphase greedy algorithms

3) SEARCHING [2 session]

a. Binary Search Trees
b. Red-Black Trees
c. Amortized Analysis
d. Splay Trees
e. Comprehensive search
    i. Backtracking
    ii. Gaming tree
    iii. Branch-and-Bound
    iv. The traveling-salesman algorithm

4) GRAPH ALGORITHMS [1 session]

f. Graph Search
g. Shortest Paths
h. Minimum Spanning Trees
i. Union-Find

5) String and pattern matching [1 session]

j. The naive string-matching algorithm
k. The Rabin-Karp algorithm
l. String matching with finite automata
m. The Knuth-Morris-Pratt algorithm

6) NP-complete problems [1 session]

n. Overview
o. Polynomial time
p. Polynomial-time verification
q. NP-completeness and reducibility
r. NP-completeness proofs
s. NP-complete problems
    i. The clique problem
    ii. The vertex-cover problem
    iii. The hamiltonian-cycle problem
    iv. The traveling-salesman problem

7) TOPOLOGICAL ALGORITHMS [1 session]

t. Geometric Graphs
u. Surfaces

v. Homology

8) **GEOMETRIC ALGORITHMS [1 session]**
   w. Plane-Sweep
   x. Delaunay Triangulations
   y. Alpha Shapes

12. Assessment Strategy

[This section details how the Learning Outcomes will be assessed. This includes formative as well as summative assessment. Where formative assessment is includes diagnostic testing, informal feedback and non-graded assessment of what a student has learned. Summative assessment is graded assessment based upon formal course work and examinations.
The assessments used should be described, the type of assessment (e.g. essay, examination, presentation, group work), along with what they aim to achieve, and at what stage in the module they take place. Include in brackets which Learning Outcomes each assessment component is designed to assess e.g. (LO1, LO2, LO3).

There will be two class exam and a final theory and practical Exam.

Class Exam I– LO1, LO2, LO3
Class Exam II – LO4, LO5, LO6
Final Theory Exam - LO1, LO2, LO3, LO4, LO5, LO6
Practical EXM – LO1, LO2, LO3, LO4, LO5, L06

13. Summary description of assessment items

[A table summarizing the assessment components of the module]

Assessment Type	Description of Item	% Weighting	Grading	Tariff	Week due
Class Exam I	Theory & Practical Exam	20%			4th week
Class Exam II	Theory & Practical Exam	20%			8th week
Final Theory Exam		40%			End of the module
Final Practical Exam		20%			End of the module
Formative Assessment		0%			Throughout the module

[Assessment Type identifies whether it is an exam, a coursework, group work, presentation etc. These can be abbreviated e.g CWK, GWK, EXM, PRS
Description of item - a very brief description of the assessment e.g. design and

write a program to manage a bank account
Weighting – the percentage of the module's total assessment the particular assessment accounts for.
Grading – the grading structure e.g. pass, merit, distinction along with the number of marks or percentages required to achieve those grades.
Tariff – the amount of work required from the student for that assessment e.g. 1000 – 1500 words for an essay or 3 hours for an exam, 15 minutes for a presentation. The tariffs should aim to be commensurate across the department.
Week due – the week in the semester the course work is to be handed in or the exam or test takes place.]

14. Learning Session Structure

[Describe the structure of each weekly learning session, e.g a 1 hour lecture followed by 30 minute tutorial, and 1 ½ hour practical workshop in a computer lab]
30 minutes Tutorial
2 hour lecture
1 ½ hour lab session

15. Learning and Teaching Methods

[Describe the range of teaching methods used e.g. lectures, tutorials, workshops, set readings. Then describe how much time each will take each week and in total for the semester e.g. one hour lecture per week (total 14 hours) and what that teaching method will be used for on the module e.g. the tutorial will consist of a set questions put to the students to informally assess their understanding of the content of the lecture, to allow them to think about and solve example problems related to the lecture content, to express their understanding in English, and to correct any misunderstanding or gaps in their knowledge of the lecture's content.]

30 minutes Tutorial – $6^{1/2}$ hours for a semester.
2 hour lecture – 22 hours for a semester.
1 ½ hour lab session – $28^{1/2}$ hours for a semester.

16. Scheme of Work

Week	Delivery Method	Content	Learning Materials	Learning Outcomes	Form of Assessment
1	Lecture, Tutorial	**INTRODUCTION** a. Review of basic data structures (Queue, Stack, link lists, trees) b. Review of basic algorithms c. Recurrences.	Slides, Notes, Exercises	Lo1	Formative Assessment
1	Lab		Exercises		

2	Lecture, Tutorial	**Design techniques** a. Divide-and-Conquer i. Multiplication ii. Recurrence relations iii. Mergesort iv. Medians v. Matrix multiplication	Slides, Notes, Exercises	LO2	Formative Assessment
2	Lab		Exercises		
3	Lecture, Tutorial	**b. Prune-and-Search** c. Dynamic Programming i. Shortest paths ii. negative cycles iii. matrix chain multiplications iv. sequence alignment v. RNA secondary structure	Slides, Notes, Exercises	LO2, LO3	Formative Assessment
3	Lab		Exercises		
4	Lecture, Tutorial	**d. Greedy Algorithms** i. Minimum spanning tree ii. Union-Find algorithms iii. Kruskal's Algorithm iv. Clustering v. Huffman Codes vi. Multiphase greedy algorithms	Slides, Notes, Exercises	LO2,LO 3	Formative Assessment
4	Lab		Exercises		
5	Lecture, Tutorial	**SEARCHING** a. Binary Search Trees b. Red-Black Trees	Slides, Notes, Exercises	LO3, LO4	Formative Assessment

		c. Amortized Analysis d. Splay Trees e. Comprehensive search			
5	Lab		Exercises		
6	Lecture, Tutorial	**SEARCHING** i. Backtracking ii. Gaming tree iii. Branch-and-Bound iv. The traveling-salesman algorithm	Slides, Notes, Exercises	LO3,LO 4, LO5	Formative Assessment
6	Lab		Exercises		
7	Lecture, Tutorial	**GRAPH ALGORITHMS** f. Graph Search g. Shortest Paths h. Minimum Spanning Trees i. Union-Find		LO35, LO6	Formative Assessment
7	Lab		Exercises		Summative Assessment
8	Lecture, Tutorial	**String and pattern matching** j. The naive string-matching algorithm k. The Rabin-Karp algorithm l. String matching with finite automata m. The Knuth-Morris-Pratt algorithm	Slides, Notes, Exercises	LO5, LO6	Formative Assessment
8	Lab		Exercises		
9	Lecture, Tutorial	**NP-complete problems** n. Overview o. Polynomial time p. Polynomial-time verification q. NP-completeness and reducibility r. NP-completeness	Slides, Notes, Exercises	LO3, LO5, LO6	Formative Assessment

		proofs s. NP-complete problems i. The clique problem ii. The vertex-cover problem iii. The hamiltonian-cycle problem iv. The traveling-salesman problem			
9	Lab		Exercises		
10	Lecture, Tutorial	**TOPOLOGICAL ALGORITHMS** t. Geometric Graphs u. Surfaces v. Homology	Slides, Notes, Exercises	LO5, LO6	Formative Assessment
10	Lab		Exercises		
11	Lecture, Tutorial	**GEOMETRIC ALGORITHMS** w. Plane-Sweep x. Delaunay Triangulations y. Alpha Shapes	Slides, Notes, Exercises	LO5, LO6	Formative Assessment
11	Lab and Review		Exercises		

17. Bibliography
[List of books or articles to be used in the module]

Levitin, A. (2007). Introduction to the design & analysis of algorithms. Boston: Pearson Addison-Wesley.

Kleinberg, J. and Tardos, E. (2006). Algorithm design. Boston: Pearson/Addison-Wesley.

Skiena, S. (1998). The algorithm design manual. Santa Clara, CA: TELOS--the

Electronic Library of Science.

Sedgewick, R. and Wayne, K. (2011). Algorithms. Upper Saddle River, NJ: Addison-Wesley.

18. Authored by
[The member of staff who wrote the modules specification. Usually the module leader, if being delivered for the first time. Include the date it was completed.]

Thenraja Vettivelraj

19. Validated and Verified by
[Another member of staff who has checked the module specification to ensure that it meets the requirements of the course of which it a part and has checked the specification for any errors. This will include the date it was approved.]