Mathematics 5365 (Fall 2018)
Computer Literacy and Programming II
Design, Analysis, and Implementation of Algorithms

Instructor: Dmitri Pavlov, Assistant Professor

Lectures: TuTh 12:30–2, MA 113

Office hours: TuTh 3:30–5 (no colloquium) or 5–6:30 (colloquium), MA 117C (few students) or 113 (many students)

Midterms: September 27 and November 1

Final exam: Saturday, December 8, 1:30–4

Credit hours: 3

Website: https://dmitripavlov.org/

Email: dmitri.pavlov@ttu.edu

1 Course Outline

A classical array of topics will be covered, including the following:

- methods: loop invariants, asymptotics, amortized analysis;
- data structures: various kinds of lists, stacks, heaps, binary search trees, memory pools, disjoint sets, stacks and queues, trees, tries, hash tables;
- sorting, medians, order statistics, priority queues;
- dynamic programming: knapsack, traveling salesman;
- divide-and-conquer algorithms: powers via squaring, binary search, merge sort, closest pair of points;
- string algorithms: Knuth–Morris–Pratt, Aho–Corasick, Thompson, suffix trees, lowest common ancestors;
- computational geometry: sweeping line, intersection of segments, convex hulls, triangulations of polygons, Voronoi diagrams and Delaunay triangulations;
- number theory: Euclid's algorithm, quadratic nonresidues, Miller–Rabin;
- graphs: search, Dijkstra, Floyd–Warshall, shortests paths, matchings, assignment problem, maximum flow, spanning trees.
- techniques: greedy algorithms, backtracking, linear programming;

2 Text

The material will be drawn from a variety of sources, individual texts will be indicated as the course progresses.

3 Catalog Course Description

Development of computer literacy and programming ability, algorithms and data structures, and recursion.

4 Course Purpose

This course will teach the basics of programming, including design of algorithms and data structures, proof of their correctness, computation worst-case and average-case running times, and implementation in the programming language of choice.

5 Course Specific Expected Learning Outcomes

Upon completion of this course, students will be able to design algorithms and data structures, prove their correctness, compute their worst-case and average-case running time, and implement them in the programming language of choice.

6 Assessment of Expected Learning Outcomes

Assignments will be given throughout the course. Each assignment will consist of a theoretical part, which requires the student to design an algorithm or a data structure, prove its correctness, compute its worst-case and average-case running time, and a practical part, which requires the student to implement this algorithm as a computer program. This program will then be automatically tested by a specialized software. Upon the correct passage of all tests the program will be manually inspected by the instructor and counted as completed if no further changes are necessary.

7 Grading

At the end of the course the number of completed assignments will be divided by the number of all assignments, and the final grade will be assigned according to the following rubric:

$$[.9, 1] \mapsto \text{A}, \qquad [.8, .9) \mapsto \text{B}, \qquad [.7, .8) \mapsto \text{C}, \qquad [.6, .7) \mapsto \text{D}, \qquad [0, .6) \mapsto \text{F}.$$

8 Collaboration

Acceptable modes of collaboration: discussing problems with classmates orally or using a blackboard. Students must indicate their collaborators in their submissions.

Unacceptable modes of collaboration:

- looking at or copying from a written solution of a classmate or somebody else;
- writing down something that a collaborator told you, but you do not understand.

9 Schedule

There will be 28 class meetings on the following days:

August 28: Introduction to programming. Abstract types, variables, assignments, conditionals, loops, loop invariants, big O.

August 30: Loop invariants, time estimates.
September 4: Worst-case and average-case running time for binary search.
September 6: Examples of loop invariants. Data structures: stack.
September 11:
September 13:
September 18:
September 20:
September 25:
September 27:
October 2:
October 4:
October 9:
October 11:
October 16:
October 18:
October 23:
October 25:
October 30:
November 1:
November 6:
November 8:
November 13:
November 15:
November 20:
November 27:
November 29:
December 4:

10 Midterms

There will be midterms on September 27 and November 1.

11 Additional Information

What will not be taught in this course:

- how to use software written by other people (e.g., Maple/Mathematica/MATLAB);
- how to use software libraries written by other people (e.g., LAPACK, NumPy, pandas, STL, Boost);
- numerical methods of any kind (covered extensively by other courses in this department);
- syntax of programming languages (must be learned independently).

12 Operating Policy 34.19: Student absence for observance of religious holy day

1. "Religious holy day" means a holy day observed by a religion whose places of worship are exempt from property taxation under Texas Tax Code §11.20.
2. A student who intends to observe a religious holy day should make that intention known in writing to the instructor prior to the absence. A student who is absent from classes for the observance of a religious holy day shall be allowed to take an examination or complete an assignment scheduled for that day within a reasonable time after the absence.
3. A student who is excused under section 2 may not be penalized for the absence; however, the instructor may respond appropriately if the student fails to complete the assignment satisfactorily.

13 Operating Policy 34.22(2b): Reasonable accommodation for students with disabilities

Any student who, because of a disability, may require special arrangements in order to meet the course requirements should contact the instructor as soon as possible to make any necessary arrangements. Students should present appropriate verification from Student Disability Services during the instructor's office hours. Please note: instructors are not allowed to provide classroom accommodations to a student until appropriate verification from Student Disability Services has been provided. For additional information, please contact Student Disability Services in West Hall or call 806–742–2405.

14 Operating Policy 34.12(5): Academic dishonesty definitions

Students must understand the principles of academic integrity, and abide by them in all class and/or course work at the University. Academic Misconduct violations are outlined Part I, section B.1 of the Code of Student Conduct. If there are questions of interpretation of academic integrity policies or about what might constitute an academic integrity violation, students are responsible for seeking guidance from the faculty member teaching the course in question.

Academic misconduct includes cheating, plagiarism, collusion, falsifying academic records, misrepresenting facts, violations of published professional ethics/standards, and any act or attempted act designed to give unfair academic advantage to oneself or another student. Additional information about academic misconduct is available in the Texas Tech University Handbook in Part II, section B of the Community Policies section in the Student Handbook at http://www.depts.ttu.edu/dos/handbook/.