CSCI 406: Algorithms Syllabus

Professors Estelle Smith & Alexandra Chakarov

Administrative Information

Instructors	Estelle Smith & Alexandra (Alex) Chakarov
Estelle Office Hours: *Alex Office Hours:*	Mon & Wed @ 2:00-3:00pm (Brown 280-G) Mon @ 9:00-10:30, Wed, Fri @ 3:00-4:00pm (CTLM 246B) or by appointment
Email	estellesmith@mines.edu, alexandra.chakarov@mines.edu
TAs	Kelly Dance (kdance), Audrey Haas (ahaas1), Sander Schott (sschott), Jayden Pahukula (jaydenpahukula), Stone Amsbaugh (samsbaugh), Matthew Desaulniers (mdesaulniers) Grant Dibala (gdibala), Nathan George (nathan_george) Nathan Webster (nwebster), Isaac Williams (idwilliams)
TA office hours	TBD
Course Web Page	Canvas Page for Algorithms
Textbook	Algorithm Design Manual, Skiena (accessible via Course Readings on Canvas)

Prerequisites

The Spring 2025 version of CSCI 406 is designed for students who have either taken the 261-262 or 200-220 sequences. CSCI 358 must also be completed prior to taking CSCI 406.

About the Course

CSCI 406 is a course that will especially appeal to those who enjoy problem-solving, logical reasoning, and tackling challenges often found in programming competitions. The concepts covered here are frequently featured in technical interviews at top software companies, as they highlight your ability to think systematically and analytically. As one former student put it, this class is about *"making computer scientists out of programmers."* However, computer science is a diverse field with room for many strengths, so we've designed this course to go beyond the traditional

"homework + exam" structure. You'll also engage in hands-on projects, grading interviews, and real-world applications to ensure a well-rounded learning experience.

Expectations and Philosophy

1. We consider Algorithms to be a moderately rigorous class in the CS curriculum. Some of you will find the class quite challenging - we make no apologies because it's our job as Mines faculty to set high standards. However, we are also there to help you succeed. Accordingly, please ensure that you seek help when you need it. Make use of class-time, office hours and Ed Discussion!

2. We understand that Algorithms is not the only course you will be taking this semester and that there will be other courses and activities that will compete for your time. It is your responsibility to ensure that you maintain a reasonable portfolio of activities (not too many, not too few!) and that you practice good time management. Specifically, please start thinking about and working on your homeworks and projects soon after they become available.

3. Cell phones are a source of distraction for everyone - accordingly please silence and place them face down. If you absolutely need to send a text or take a call, please leave the classroom for the duration of the activity. Tablets, laptops, etc., can also similarly be a source of distraction and should be put away unless you are using them for activities related to class.

4. Please review important policies at the end of this document.

Delivery and Organization

1. The class will be delivered in person with traditional lectures on Mondays, Wednesdays, and Fridays. (We may provide optional video recordings and content in Canvas for purposes of pre-learning before class or review after class.) There may be more opportunities for interaction on many Fridays, when we may provide additional opportunities for problem solving, coding demos, or homework/project hints.

2. In-class attendance is *always* required MWF. Paper worksheets will be handed out at the start of class to support your active learning during lectures; please bring a writing utensil to class to work on these problems. Paper worksheets will **not** be turned in; please retain your worksheets for review ahead of exams, as some exam questions may closely resemble in-class worksheet problems. Instead of collecting worksheets, **instructors will use iClicker questions to record attendance** and collect answers about the worksheets; you must bring an Internet-enabled device such as a laptop or smartphone that can be used to answer iClicker questions. Your attendance score is based on completion (i.e., *not* accuracy) of iClicker questions.

3. Five (5) grace days will be allowed for attendance. You may skip up to 5 days of in-class attendance, no questions asked, without incurring a penalty to your grade for the course. Do *not* email the instructors or submit a request on EdDiscussion if you would like to use a grace

day. All grace days will be applied automatically at the end of the semester. The intent of this provision is to allow for some flexibility for personal days involving travel, work, or other obligations that are not suitable for an excused absence. If you have an excused absence related to, e.g., illness, family emergency, etc., you should submit an excused absence request instead.

4. On Canvas you will find the class organized by Weeks. Each week ends on Saturday at midnight. Deliverables will usually be due at this time, with a few notable exceptions such as Homework 3, AlgoBOWL and intermediate Project submissions.

5. YouTube videos recorded during the pandemic will be made available. Professor Alex Chakarov will also record her lectures and post them to Canvas following class. Neither of these resources are a substitute for in-class attendance, but they may be useful in the event you are absent in class or for review.

6. Homework is required to be submitted to Gradescope as a PDF. We strongly recommend that you use LaTeX/Overleaf for written homework and project submissions. LaTeX documents can be compiled into PDF for submission to Gradescope.

7. Please use Ed Discussion for online discussion and questions.

Semester at a Glance

Week #	Content	Major Deliverables
Week 1 (ends Jan 11)	Unweighted Graphs	Syllabus quiz, DFS/BFS Review Problem
Week 2 (ends Jan 18)	Shortest Paths	HW1
Week 3 (ends Jan 25)	Advanced Heaps	HW2
Week 4 (ends Feb 1)	MSTs & Disjoint Sets	Maze Project
Week 5 (ends Feb 8)	Network Flows	HW3 (2/05/25); **Exam 1 (Fri, 2/7/25)**
Week 6 (ends Feb 15)	Bipartite Matching	HW4; Maze Grading Interviews
Week 7 (ends Feb 22)	Parallel Algorithms	HW5; Maze Grading Interviews
Week 8 (ends March 1)	Review	AlgoBOWL Week
Week 9 (ends March 8)	Dynamic Programming	**Exam 2 (Mon, 3/3/25)**
Week 10 (ends March 15)	Dynamic Programming	HW6
Spring Break	**March 15-23**	No class or assignments :)
Week 11 (ends March 29)	Dynamic Programming	HW7 Part 1; DP Project Part 1
Week 12 (ends April 5)	NPC - Reductions	HW7 Part 2; DP Project Part 2
Week 13 (ends April 12)	NPC - Definitions	
Week 14 (ends April 19)	NPC - More Reductions	HW8
Week 15 (ends April 26)	Coping with NPC	HW9
Week 16 (ends April 30)	Review	
Finals Week (May 2-7)	**Time & Room TBD**	**Cumulative, emphasis on weeks 9-16**

Grading Rubric at a Glance

Projects	30%
Two Exams	20% (10% each)
Weekly Homeworks	20%
Attendance	10%
Final Exam	20%

Grading Details

1. **Projects:** There will be three projects, each worth 10%. (a) Maze Project (individual) (b) AlgoBOWL project (group) (c) Dynamic Programming Project (individual). Late Policy For Maze and DP: TBD. AlgoBOWL consists of multiple mini-deadlines. Your group **must** meet all AlgoBOWL without exception.

2. **Homework Assignments (individual):** 9 HW submissions, cumulatively worth 20%. We will automatically discard the one HW with the lowest score. Late HWs will **not** be accepted. It is your responsibility to leave margin for any Gradescope submission issues. You may collaborate on HWs with your classmates (anyone taking the class this semester), but you must list the names of students with whom you worked. Do not consult the internet or Generative AI for solutions - this has been the source of academic integrity violations in recent semesters.

3. **Attendance iClicker Questions:** Approximately 40 days of attendance questions, based on paper worksheets distributed in-class. A good-faith effort is expected; however, attendance scores are based on participation rather than accuracy. You must attend a minimum of 70% of class periods, or you risk failing the class. Five (5) grace days (absences) are allowed, no questions asked. You may equivalently attend either section A or section B to answer iClicker questions and earn your participation score for the day.

Grading Policies and Procedures

1. *Submission Logistics:* Typeset PDFs of HWs must be uploaded on Gradescope. These will **not** be accepted by email. Project code for individual projects (Maze and Dynamic Programming) must be written in Python and submitted to the autograder. Project code for AlgoBOWL will *not* be turned in, and can be written in any desired coding language agreed upon by your group. All projects will additionally involve *either* grading interviews *or* submission of typeset written PDF reports. Maze Project will use **grading interviews** (i.e. 20-30 minute, scheduled conversations with a TA to answer questions related to the project problem and code). AlgoBOWL and DP are TBD.

2. *Extensions:* Extensions on deadlines will be given **only** if there are extenuating circumstances. Requests for extensions must be made during business hours **before** the deadline (unless you were physically unable to do so due to a documented accident or injury, in which case

proof must be furnished). Requests must be made by submitting an Extension Request on EdDiscussion. Since most assignments are due Saturday at midnight, this implies that most requests must be made no later than 5pm on Fridays: Do NOT expect replies to emails or EdDiscussion requests over the weekend. Therefore, be responsible, professional, and proactive in seeking support on your HW questions or in requesting extensions.

3. *Regrades*: Regrade requests must be received within 7 days after grades are made available. A thoughtful regrade request must explain (a) what rubric item was incorrectly applied/not applied and (b) why you think so (unless it's obvious, like a T/F). Otherwise, the regrade request may be rejected as "malformed." Our general policy is that regrades will not result in lower grades. However, multiple instances of malformed requests from the same student could result in a lower grade (after a suitable warning). If you're seeking clarification on a HW question, we would encourage you to first post a question (preferably public) on EdDiscussion.

4. *Final grades*: The scores listed in the table below will guarantee the corresponding grades.

93.00	90.00	87.00	83.00	80.00	77.00	73.00	70.00	67.00	63.00	60.00
A	A−	B+	B	B−	C+	C	C−	D+	D	D−

5. To pass the class, you must have a passing total score **AND** a passing score in the project component of the class **AND** a passing score in **two** out of the three exams.

Learning Objectives

The Learning Objectives below are classified using Webb's Depth of Knowledge (DoK). In increasing order, these are Level 1 (recall), Level 2 (skill/concept), Level 3 (strategic thinking) and Level 4 (extended thinking). See DoK. At the end of the semester, you should be able to:

1. Solve math problems related to algorithm complexity and correctness by utilizing discrete math techniques (e.g., logic, asymptotics, combinatorics, probability). Levels 1, 2.
2. Execute by hand any algorithm listed in the syllabus on example inputs without referring to a written description of the algorithm. Levels 1, 2.
3. Determine the theoretical (asymptotic) time complexity of a given algorithm. Level 2.
4. Implement (i.e., code) and compare performance of different algorithms for the same problem; semi-quantitatively match actual run times to theoretical time complexities. Level 3.
5. Design an efficient algorithm to solve a given problem, analyze its theoretical time and space complexity and prove its correctness. Specifically
   (a) Design and implement dynamic programing solutions to solve optimization problems. Levels 3, 4.
   (b) Formulate and implement the solution to a non-trivial maze problem by modeling as a graph. Level 3.
6. Develop a competitive but necessarily suboptimal ("quick and dirty") solution for an NPC problem. Level 3, 4.

7. Define the complexity classes P, NP and NPC and explain the implications of a problem belonging to each class; apply the concept of a *reduction* to simple examples. Levels 1, 2.
8. Function effectively as a team member on an algorithm design and implementation team with effectiveness being determined by peer ratings, self-assessment and instructor observations. Level 3,4.
9. Demonstrate precision and attention to detail with respect to definitions, terminology, notation and language as measured by performance on multiple-choice and true-false questions on online-quizzes and exams. Levels 1, 2.

Class Policies

Collaboration Policy

1. If the project is an individual effort project, you are not allowed to give code you have developed to another student or use code provided by another student. If the project is a group project, you are only allowed to share code with your group members.
2. You are encouraged to discuss programming assignments with other students in the class, as long as the following rules are followed: a. You view another student's code only for the purpose of offering/receiving debugging assistance. Students can only give advice on what problems to look for; they cannot debug your code for you. All changes to your code must be made by you. b. Your discussion is subject to the empty hands policy, which means you leave the discussion without any record [electronic, mechanical or otherwise] of the discussion.
3. Any material from any outside source such as books, projects, and in particular, from the Web, should be properly referenced and should only be used if specifically allowed for the assignment.
4. To prevent unintended sharing, any code stored in a hosted repository (e.g., on github) must be private. For group projects, your team members may, of course, be collaborators.
5. If you are aware of students violating this policy, you are encouraged to inform the professor of the course. Violating this policy will be treated as an academic misconduct for all students involved.

0.1 Generative AI Policy

CSCI 406 uses the **GENERAL PROHIBITION** policy for Generative AI as outlined by the Office of Academic Affairs.

By submitting work for evaluation in this course, you represent it as your own intellectual product. Submitting content for evaluation (e.g., ideas, text, code, images) that was generated, in whole or in part, by Generative Artificial Intelligence tools (including, but not limited to, ChatGPT and other large language models) would be considered Academic Misconduct in violation of Mines Academic Integrity/Misconduct Policy.

Learning Environment

Fundamentally, I expect and require respect in this course for yourself, your classmates, and your instructor and TAs.

1. Respect for yourself includes taking care of yourself physically and mentally and advocating for an environment that facilitates learning for you.

2. Respect for your classmates includes recognizing and appreciating the diversity of backgrounds and experiences of your classmates and making it your interest to foster a learning environment for everyone; all are welcome.
3. Respect for your instructors (as well as your classmates) includes not participating in disruptive or distracting behavior: talking, playing games, or web surfing during lecture, for instance, make it difficult for others to focus on the reason we are all here.
4. Respect must be mutual to be effective; we (your instructors) and your TAs will be held to the same standards of respect.

Please let your instructor know if you become aware of an issue with the classroom (or out-of-classroom) environment with regards to these policies.