CS 5200 ALGORITHMS
Fall 2019

George Markowsky
Computer Science Department
Missouri S&T

1 Course Description

COMP SCI 5200 Analysis Of Algorithms (LEC 3.0)
The purpose of this course is to teach the techniques needed to analyze algorithms. The focus of the presentation is on the practical application of these techniques to such as sorting, backtracking, and graph algorithms. Prerequisite: A "C" or better grade in Comp Sci 2500.

2 Goals for CS 5200

After completing this course, the student should be able to do the following.

1. The student must be able to write a correct inductive proof.
2. The student should be aware of the different classes of algorithm design strategies and be able to come up with prototypical algorithms from each class.
3. The student should have broad familiarity with some of the most important algorithms discovered so far, and understand how to use them as prototypes for new algorithms or how to adapt them to different situations.
4. When presented with a problem and an algorithm which claims to solve the problem, the student should be able to determine whether the algorithm does indeed solve the problem. Ideally, the student should be able to give a correct proof that the algorithm does indeed solve the problem. If the student cannot give a proof, the student should be able to identify the cases for which the proof does not work.
5. When presented with an algorithm, the student should be able to analyze the performance of the algorithm. Ideally, this could be precise, but if a very detailed analysis is not possible the student should at least be able

to determine whether the algorithm is exponential, polynomial (degree?), linear or sublinear. When presented with a problem, the student should be able to suggest one or more algorithms for solving the problem. The student should be able to create a table comparing the algorithms to one another so that the trade-offs are apparent.

6. When presented with a problem, the student should be able to suggest one or more algorithms for solving the problem. The student should be able to create a table comparing the algorithms to one another so that the trade-offs are apparent.

7. The student should understand how to write programs that carry out an algorithm.

8. The student should understand how to test programs to determine that they accurately carry out the algorithm.

3 Prerequisites

A "C" or better grade in Comp Sci 2500.

4 Grading

Generally, I have had a bad experience with grading homework. At the same time, I believe that there is a lot of benefit to homework if done properly. The biggest problem with homework has been the tendency of some students to copy solutions from the Internet and submit them as their own work. Clearly, they are not learning much in this way and making it difficult to assess how much they know. For this reason, the bulk of the grades will be determined by the exams, which will all be takehome exams. The dates of these exams are on the schedule shown below. The takehome exams will contain problems that are not likely to have solutions on the Internet and will be customized for each student so that no two exams will be identical. They will be equivalent in difficulty and content, but be distinct and have different answers. There will be two prelims and one final.

The relative worth of homeworks to exams will depend on how much grading help is available for the homeworks. If X is the fractional contribution of your homework grade to your final grade, (1 - X) will be the fractional contribution of your exam average to the your final grade. Each prelim will contribute 25% to your exam average and the final will contribute 50%. I will let you know the final value of X by the time of the first prelim.

To reward improvement, if you do better on a later exam, the better grade will replace all lower grades that came before it. If you do worse on subsequent exams, you get to keep the grade you originally got. Thus, improvement is rewarded. For example, if you got 50% on prelim 1, 90% on prelim 2, and 80 % on the final, your final grade will be calculated using 90% for prelim 1, 90% for

Numerical Range	Undergraduate Grade	Graduate Grade
85 or above	A	A
75-84	B	B
65-74	C	C
55-64	D	F
54 or below	F	F

Table 1: Numerical Scores Converted to Letter Grades

prelim 2, and 80% for the final. In this case, your exam average would be 85% instead of 75% which would result from just averaging the exams. This means that a poor exam score will not hurt you if you improve.

1. The grades will be assigned as shown in Table 1.
2. The homework average, hw, will just be the average of your homework grades. Unsubmitted homeworks will count as 0.
3. The exam average, ex, is computed as follows, where p1 and p2 are the two prelim scores and f is the final score.

   $$\text{ex} = .25*\max(p1,p2,f) + .25*\max(p2,f) + .50*f$$

   The final average is computed by

   $$\text{favg} = X*\text{hw} + (1 - X)*\text{ex}$$

4. All numbers are rounded and the letter grades are assigned according the scale shown in Table 1.
5. It is important that you do and understand the homework, since the exams will be based on the homework problems.

5 Additional Notes

1. Tentative dates for the exams are listed in the schedule below. These times might move around as necessary.
2. All exams are cumulative and cover all the material up to the time that they are given.
3. Partial credit will be given on exam problems.

4. I want people to work on the homework individually. You can talk to each other and give help, but this help should not take the form of letting other people copy your work. It is important that you understand how to do all the problems on your own so you can do well on the exams. If you need help, please ask questions in class, on Canvas, and come during office hours.

5. If a problem asks you to write a program, a function or a procedure, always submit a listing and output, even if the problem does not explicitly ask for these. The "official" programming language of the course will be Python 3.X, which means that all submitted programs must be in Python 3.X. Reference material for Python will be available on Canvas. Also, the programs that I will present in class will provide an introduction to Python. You will discover that it looks a lot like pseudo-code.

6. All homework submissions will be electronic via Canvas. More details will be provided with every assignment.

7. Many materials will be available via Canvas using your MST credentials. You will automatically be enrolled in the appropriate area of Canvas once your registration in the course is complete.

8. If your programs have bugs, I expect you to make a reasonable effort to find the bug on your own. I will be happy to help you find problems in your programs, but you must come with evidence that you have tried to find the problem on your own and the program I see should have evidence of your efforts to debug it.

9. If you run out of time and must turn in a program that doesn't run, submit output showing the crash and the error message along with the program.

10. Be sure that your code includes comments that explain what you are doing if it is not completely obvious. It is up to you to explain what you are doing. You will appreciate this point better once you have experience reading other people's code.

11. If you do not understand a problem get a clarification from me. Do not waste a lot of time working on something that you don't understand. You should not expect to use the fact that you did not understand a problem as an excuse for doing poor work. You should make sure that you understand each problem before you attempt to do it.

12. Homework submitted late with will lose points. Please turn in your work on time. If special circumstances will cause you to be late with your homework, please contact me as soon as you can about this.

13. Since this is an advanced course, I expect very high-quality submissions from you. Points will be deducted for sloppy or disorganized work.

14. Any program that you submit must include sample output that adequately tests it. This sample run should not be copied from the screen by hand – it should be either output generated by the program and collected by you or a screen dump if that captures the results better. Be sure to include the output/screen dump at a reasonable place in your document. You should think carefully about what constitutes an adequate test for each program that you write. You will lose points for inadequate testing.

15. When you write programs, pay attention to the human interface. The requests for data should be reasonable. Ridiculous interfaces will lose points just for being ridiculous.

16. All submissions must be typed – no handwritten submissions will be accepted.

17. Be sure to answer the question. If you are asked for an analysis of an algorithm, be sure to supply one. Do not assume that you will receive the bulk of the points simply for coding it. Also, if I ask you to analyze an algorithm, analyze the one you are given. Don't analyze some other algorithm. Don't answer questions that are "almost" like the questions you are asked.

18. I expect your algorithms to be reasonably efficient. Just simply whipping something together that gets the job done might not be enough. Also, if you make modifications to algorithms, you will lose points if you make the algorithm perform significantly less efficiently than what was presented.

19. Put some thought into organizing your submissions.

20. Submit all necessary pieces. I don't want to guess what data types you used, etc. You will lose points for submitting poorly organized and unreadable material

21. You will lose points for poor programming style. I do not want to see hoards of global variables in your programs. You have been taught how to do things correctly and I want to see you do it.

22. I am only interested in grading your original work. I am not interested in grading solutions to the problems that have been posted by other professors on the Web. You can lose many points if you simply copy solutions from other people or other sources.

23. If you have problems with this course and need help come in to see me immediately.

24. Don't postpone doing your work until the end of the semester and then try to learn all the material in a week.

6 Contact Information & Office Hours

Name	George Markowsky
Office	323 CS Building
Email	markov@mst.edu
Office Hours	11am-11:50 am MWF 1:00-1:50 pm TuTh
Phone	573-341-6138

There might be a graduate assistant or grader for this course. Details will be announced when this is known

7 Textbook

The textbook for the course is *Introduction to Algorithms: 3rd Edition* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. There will also be supplemental materials available on Canvas.

8 CS 5200 Schedule

Tables 2 and 3 contain the tentative schedule for CS 5200 in Fall 2019. I reserve the right to modify the schedule as the semester progresses.

Day	Date	Lec Num	Topic
M	19-Aug	1	Intro to Course & Python
W	21-Aug	2	Recursion I
F	23-Aug	3	Recursion II
M	26-Aug	4	Induction
W	28-Aug	5	Big Oh, Fibonacci Numbers
F	30-Aug	6	Sum, Series
M	02-Sep		LABOR DAY - NO CLASS
W	04-Sep	7	Recurrences, Master Theorem
F	06-Sep	8	Fast Multiplication, Divide & Conquer Algorithms
M	09-Sep	9	Probability I
W	11-Sep	10	Probability II
F	13-Sep	11	Probability & Cryptography
M	16-Sep		PRELIM 1 TAKEHOME - NO CLASS
W	18-Sep		PRELIM 1 TAKEHOME - NO CLASS
F	20-Sep		PRELIM 1 TAKEHOME - NO CLASS
M	23-Sep	12	Heapsort & Quicksort
W	25-Sep	13	Linear Sorting, Lower Bounds
F	27-Sep	14	Medians
M	30-Sep	15	Elementary Data Structures
W	02-Oct	16	Data Storage Hashing
F	04-Oct	17	BInary Search Trees
M	07-Oct	18	Red Black Trees
W	09-Oct	19	Augmenting Data Structures
F	11-Oct	20	Dynamic Programming I
M	14-Oct	21	Dynamic Programming II
W	16-Oct	22	Greedy Algorithms
F	18-Oct	23	Matroids
M	21-Oct		PRELIM 2 TAKEHOME - NO CLASS
W	23-Oct		PRELIM 2 TAKEHOME - NO CLASS
F	25-Oct		PRELIM 2 TAKEHOME - NO CLASS
M	28-Oct	24	Amortized Analysis I
W	30-Oct	25	Amortized Analysis II

Table 2: Class Schedule Part I

Day	Date	Lec Num	Topic
F	01-Nov	26	B-Trees
M	04-Nov	27	Binomial Heaps
W	06-Nov	28	Fibonacci Heaps
F	08-Nov	29	van Emde Boas Trees
M	11-Nov	30	Disjoint Unions
W	13-Nov	31	Minimum Spanning Tree
F	15-Nov	32	Single-Source Shortest Paths
M	18-Nov	33	All-Pairs Shortest Paths
W	20-Nov	34	Maximum Flow
F	22-Nov	35	NP-Complete Problems I
M	25-Nov		THANKSGIVING BREAK
W	27-Nov		THANKSGIVING BREAK
F	29-Nov		THANKSGIVING BREAK
M	02-Dec	36	NP-Complete Problems II
W	04-Dec	37	Approximation Algorithms
F	06-Dec		TAKEHOME FINAL – NO CLASS
M	09-Dec		TAKEHOME FINAL – EXAM WEEK
W	11-Dec		TAKEHOME FINAL – EXAM WEEK

Table 3: Class Schedule II

9 Academic Honesty

Material in this section is used by courtesy of Dr. Patrick Taylor of S&T.

You're here to learn and better yourself! Write all your work in your own words, and write your own code. Do not copy-paste (plagiarize) from any source. If you are not sure, err on the side of caution and do your work independently. Occasional infrequent help from a friend when your are really stuck may be reasonable, though if that "help" is frequent enough that your collaboration results in almost identical code, it was too much collaboration for an assignment intended to be independent work (which all are unless explicitly assigned as group work).

If you are found to be engaging in any form of academic dishonesty, the most severe penalties permitted by the university will be enacted. Incidences will typically result in grades of 0 for the respective course components, as well as notification of the student's advisor, the student's department chair, and the campus undergraduate studies office. Further academic sanctions may be imposed as well in accordance with university regulations (`http://academicsupport.mst.`

edu/academicintegrity/). Those who allow others to copy their work are also committing plagiarism and will be subjected to the same procedures.

The Honor Code can be found at this link: http://stuco.mst.edu/honor-code/. Page 30 of the Student Academic Regulations handbook describes the student standard of conduct relative to the University of Missouri System's Collected Rules and Regulations section 200.010, and offers descriptions of academic dishonesty including cheating, plagiarism or sabotage (http://registrar.mst.edu/academicregs/index.html). Also see: http://academicsupport.mst.edu/academicintegrity/studentresources-ai

We check your assignments against each other with software that is VERY good at detecting similarities and differences between any text files, including your source files. These methods are difficult, if not impossible to trick. Please do not try to copy-paste, share sources directly, or write all your code or documents in a group or pair for individual assignments.

Attempting to deceive attendance checking procedures is considered academic dishonesty for ALL parties involved. For example, do not submit someone else's pre-lab or lab assignment for them because they are not attending class.

10 Office of Academic Support

The following is the contact information for the Office of Academic Support, which oversees academic integrity among other things.

Jeff Cawlfield, Vice Provost
Office of Academic Support 105 Norwood Hall
320 West 12th Street
Rolla, MO 65409-1520
Phone: 573-341-7276
Email: ugs@mst.edu
Web: http://ugs.mst.edu

The material in the remaining sections of this syllabus is based on the material produced by that office.

11 Title IX

Missouri University of Science and Technology is committed to the safety and well-being of all members of its community. US Federal Law Title IX states that no member of the university community shall, on the basis of sex, be excluded from participation in, or be denied benefits of, or be subjected to discrimination under any education program or activity. Furthermore, in accordance with Title IX guidelines from the US Office of Civil Rights, Missouri S&T requires that all faculty and staff members report, to the Missouri S&T Title IX Coordinator, any notice of sexual harassment, abuse, and/or violence (including personal

relational abuse, relational/domestic violence, and stalking) disclosed through communication including but not limited to direct conversation, email, social media, classroom papers and homework exercises.

Missouri S&T's Title IX Coordinator is Chief Diversity Officer Neil Outar. Contact him (naoutar@mst.edu; (573) 341-6038; 203 Centennial Hall) to report Title IX violations. To learn more about Title IX resources and reporting options (confidential and non-confidential) available to Missouri S&T students, staff, and faculty, please visit `http://titleix.mst.edu`.

To learn more about Title IX resources and reporting options (confidential and non-confidential) available to Missouri S&T students, staff, and faculty, please visit `http://titleix.mst.edu`.

12 Student Honor Code and Academic Integrity

Academic integrity is very important in ALL S&T classes. Academic integrity matters to you, it matters to the discipline of Computer Science and it matters to future employers. You must pay attention to the Honor Code developed and endorsed by the Missouri S&T Student Council: the Honor Code can be found at `http://stuco.mst.edu/honor-code`.

I encourage you to read and reflect upon the Honor code and its emphasis on HONESTY and RESPECT. Page 30 of the Student Academic Regulations handbook describes the student standard of conduct relative to the University of Missouri System's Collected Rules and Regulations section 200.010, and offers descriptions of academic dishonesty including cheating, plagiarism or sabotage `http://registrar.mst.edu/academicregs/index.html`. Additional guidance for faculty, including the University's Academic Dishonesty Procedures, is available on-line at `http://academicsupport.mst.edu`. Other informational resources for students regarding ethics and integrity can be found online at `http://academicsupport.mst.edu/academicintegrity/studentresources-ai`.

13 S&Tconnect

S&TConnect can be found at `https://canvas.mst.edu/`. The S&Tconnect icon is on the left toolbar. S&Tconnect provides an enhanced system that allows students to request appointments with their instructors and advisors via the S&Tconnect calendar, which syncs with the faculty or staff member's Outlook Exchange calendar. S&Tconnect will also facilitate better communication overall to help build student academic success and increase student retention. S&Tconnect Early Alert has replaced the Academic Alert system used by Missouri S&T. If training is needed, please contact Rachel Morris at rachelm@mst.edu or 341-7600. If you want to reach George Markowsky, please go through the CS Department staff in 325 CS Building.

14 Classroom Egress Maps

Students should familiarize themselves with the classroom egress maps posted on-line at `http://designconstruction.mst.edu/floorplan`.

15 Accessibility and Accommodations

It is the university's goal that learning experiences be as accessible as possible. If you anticipate or experience physical or academic barriers based on disability, please contact Disability Support Services at (573) 3416655, dss@mst.edu, or visit `http://dss.mst.edu` for information, or go to `https://mineraccess.mst.edu` to initiate the accommodation process.

Please be aware that any accessible tables and chairs in this room should remain available for students who find that standard classroom seating is not usable.

16 LEAD Learning Assistance

The Learning Enhancement Across Disciplines Program (LEAD) sponsors free learning assistance in a wide range of courses for students who wish to increase their understanding, improve their skills, and validate their mastery of concepts and content in order to achieve their full potential. LEAD assistance starts no later than the third week of classes. Check out the online schedule at `http://lead.mst.edu/assist`, using zoom buttons to enlarge the view. Look to see what courses you are taking have collaborative LEAD learning centers (bottom half of schedule) and/or Individualized LEAD tutoring (top half of the schedule). For more information, contact the LEAD office at 341-7276 or email lead@mst.edu or visit `http://lead.mst.edu`.

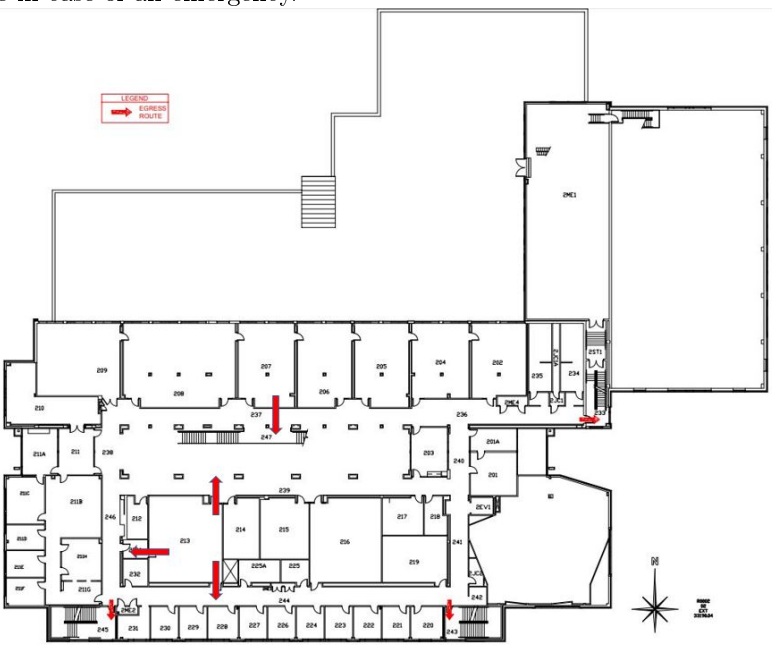
17 The Burns & McDonnell Student Success Center

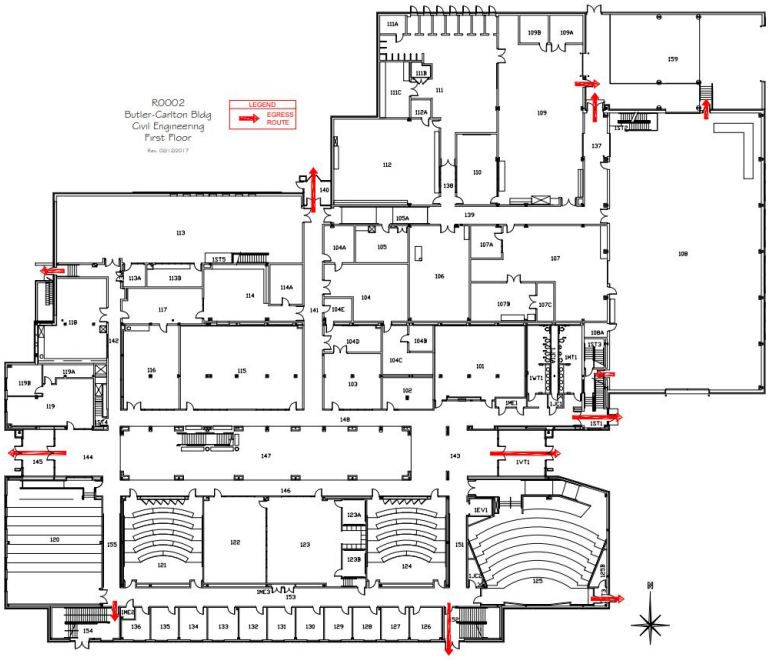
The Student Success Center is a centralized location designed for students to visit and feel comfortable about utilizing the campus resources available. The Student Success Center was developed as a campus wide initiative to foster a sense of responsibility and self-directedness to all S&T students by providing peer mentors, caring staff, and approachable faculty and administrators who are student centered and supportive of student success. Visit the B&MSSC at 198 Toomey Hall; 573-341-7596; success@mst.edu; facebook: `www.facebook.com/SandTssc` ; web: `http://studentsuccess.mst.edu/`.

If you have any questions about the information listed above, please contact the Office of Academic Support at 573-341-7276.

18 Classroom Egress Maps

Below is the map of the second floor of the Butler-Carlton building. You should review it and note possible escape routes in the event of a fire or other emergency. There are three doors to our classroom (213) and once you get out there are a number of staircases that you can use to reach the first floor. The first floor map is also shown below. Please review these diagrams carefully – they could save your life in case of an emergency.





You can find more maps of the Butler-Carlton building and other buildings at `http://designconstruction.mst.edu/floorplan/`.