DEPARTMENT: Mathematics
COURSE: MAT 2540
TITLE: Discrete Structures and Algorithms II
DESCRIPTION: Topics include predicate logic, recurrence relations, graphs, trees, digital logic, computational complexity and elementary computability.


CREDITS: 3 (2 class hours, 2 lab hours)

PREREQUISITES: MAT 2440; pre- or corequisite: CST 3503

Prepared by Professors:
J. Natov
S. Singh
A. Taraporevala
Fall 2015

A. Testing Guidelines:
The following exams should be scheduled:
1. A one-hour exam at the end of the First Quarter
2. A one-session exam at the end of the Second Quarter
3. A one-hour exam at the end of the Third Quarter
4. A one-session Final Examination

B. A Computer Algebra System will be used in class and for projects.
Learning Outcomes
for
MAT 2540 Discrete Structures and Algorithms II

1. Students will study the efficiency of algorithms.

2. Students will be able to compare data structures and algorithms.

3. Students will be able to use the master theorem to solve recurrences that arise from divide-and-conquer algorithms.

4. Students will be able to use computer technology to assist in the above.
Mathematics Department Policy on Lateness/Absence

A student may be absent during the semester without penalty for 10% of the class instructional sessions. Therefore,

If the class meets:                      The allowable absence is:

1 time per week                       2 absences per semester
2 times per week                      3 absences per semester

Students who have been excessively absent and failed the course at the end of the semester will receive either

the WU grade if they have attended the course at least once. This includes students who stop attending without officially withdrawing from the course.

the WN grade if they have never attended the course.

In credit bearing courses, the WU and WN grades count as an F in the computation of the GPA. While WU and WN grades in non-credit developmental courses do not count in the GPA, the WU grade does count toward the limit of 2 attempts for a developmental course.

The official Mathematics Department policy is that two lateness's (this includes arriving late or leaving early) is equivalent to one absence.

Every withdrawal (official or unofficial) can affect a student’s financial aid status, because withdrawal from a course will change the number of credits or equated credits that are counted toward financial aid.

New York City College of Technology Policy on Academic Integrity

Students and all others who work with information, ideas, texts, images, music, inventions, and other intellectual property owe their audience and sources accuracy and honesty in using, crediting, and citing sources. As a community of intellectual and professional workers, the College recognizes its responsibility for providing instruction in information literacy and academic integrity, offering models of good practice, and responding vigilantly and appropriately to infractions of academic integrity. Accordingly, academic dishonesty is prohibited in The City University of New York and at New York City College of Technology and is punishable by penalties, including failing grades, suspension, and expulsion. The complete text of the College policy on Academic Integrity may be found in the catalog.
<table>
<thead>
<tr>
<th>Week</th>
<th>Discrete Structures and Algorithms II</th>
<th>Homework</th>
</tr>
</thead>
</table>
| 1-5   | 3.1 Algorithms *pages* 191-200  
3.2 The Growth of functions pages 204-216  
3.3 Complexity of algorithms pages 218-228  
**First Examination**                | **P. 202**: 3-27 odd  
P. 216: 1-27 odd, 34-42 all  
P. 229: 1-5 all  
**Project 1**: Generate random lists of numbers of size n, and compare the running times of sort (or search) algorithms. Compile the data and compare to the expected running time from big O. |
| 6-8   | 11.2 Applications of Trees *pages* 757 – 769  
11.4 Spanning Trees *pages* 785 – 795  
11.5 Minimum Spanning Trees *pages* 797 – 802  
**Mid –semester Examination**               | **P. 769**: 1-7 odd, 11, 19, 21, 22, 37  
P. 795: 2 – 6 all, 13 – 15 all, 16, 27, 28, 30  
P. 802: 1, 2, 3, 6, 7  
**Project 2**: Create a tree using some computer application and write the code needed to search it |
| 9-12  | 8.1 Applications of Recurrence Relations *pages* 501 – 510  
8.2 Solving Linear Recurrence Relations *pages* 514 - 524  
8.3 Divide and Conquer Algorithms and Recurrence relations: pages 527-532  
8.4(Optional) Generating Functions: pages 537-548  
**Third Examination**               | **P. 510**: 13, 16, 28, 31, 33  
P. 524: 1, 3, 11, 17, 23, 25, 27  
P. 535: 1-21 odd  
P. 549: 1-11 odd. |
| 13-14 | 5.3 Recursive Definitions and Structural Induction *pages* 344 – 357  
5.4 Recursive Algorithms pages 360 – 370               | **P. 357**: 1, 3, 5, 7, 30, 33 - 35 all, 43, 44, MATLAB definition of Ackermann’s function, 48*, 51*, 60, 61  
P. 370: 1 – 5 odd, 7 - 10 all, 15, 16, 29, 30, 37*, 46, 50 - 52 all |
| 14-15 | **Final Examination Review and Finals**                   |                                                                           |