

General Instructions

1. Write the theory and programming assignments on different pages. **Indicate at the top your names, ID numbers and e-mails.**
2. Assignments can be done in pairs.
3. The box for submitting exercises is in the Schreiber building, at the basement floor. Directly to your left, there are several large boxes; the course's box is the leftmost. Exercises will be returned in the "hive" next to the large boxes; the course's cell is in the rightmost column, near the top.

Theoretical Assignments

1. Follow the exercise specifications exactly. Seek clarifications if needed.
2. Your proofs should be both rigorous and clear.

Programming Assignments

1. **Make sure your names, ID numbers and e-mails are printed on the report.** If the checker encounters a technical problem and has no way to contact you, **this will result in a significant point reduction.**
2. Follow the exercise specifications exactly.
3. There is no need to print the code.
4. Your code should run properly on Python 2.7.6 on linux (it doesn't have to run on Python 3 and above). It is required that it should run on `nova.cs.tau.ac.il` using `/usr/bin/python`. Test it before submitting it!
5. The one exception to the rule above is plots. You are not required to create your plots using Python on nova (although it is possible, using the `savefig` command from `pylab`). If you choose not to, then make sure your submission saves all the data required to generate the plot in a separate file, one file per plot.

6. For each programming assignment: In the printed submitted material indicate the path to the directory from which the program is executed. Make sure that the directory and files are read accessible by everyone (the output location of created/modified files should be a parameter to the program). This could be in your personal home directory, or elsewhere, as long as it is accessible.
7. The directory should contain a README file that has the student(s) name(s) and following detailed:
 - (a) A description of each relevant source/input file
 - (b) A command to execute and its parameters if any. This command should also set any required environment. You can specify a different command per question or subquestion, as long as it is clear, and there are no dependencies between the commands (i.e., command #2 does not assume command #1 was run). If you need such a dependency - make one command for two questions, etc.
 - (c) A description of each relevant output file
8. Your program should not output warnings or errors when run correctly.
9. Make sure your program remains viable if your submitted files are moved (for example, don't refer to files you do not submit, and paths that are not necessarily accessible to others). In particular, watch for OS-dependent names (e.g., Windows-type path names).
10. Your code must be clear. Use meaningful variable names and sufficient comments to achieve this.
11. Your plots should be clear. Set meaningful labels and ranges on the axes, as well as meaningful titles and legends.
12. It is recommended that you use `numpy` vector and matrix operations instead of for loops with many iterations, whenever possible. This makes for faster, more compact programs.