Math 190 **Topics: Applied Algebra** **Spring 2024**

Course Information

TIME: E+ block (Mo, We 10:30–11:45)
LOCATION: see SIS (this is an in-person class)
INSTRUCTOR: George McNinch
PREREQUISITES/RECOMMENDED: Math 70 (Linear Algebra), Math 145 (Abstract Algebra)
QUESTIONS: please e-mail `george.mcninch@tufts.edu`
OFFICE: 559 JCC

COURSE DESCRIPTION:

The course will include several *modules* or *components* of content in the course; the course will spend roughly 4 weeks (8 lectures) on each component.

- **representations of finite groups**. After formulating/recalling basic notions and examples of finite groups G, we will investigate representations; a representation of G is a homomorphism $G \to \mathrm{GL}_n(\mathbf{C})$ from G to the group of invertible $n \times n$ matrices.

  The *characters* or (certain) representations provide an alternate basis of the vector space of (all) functions on G which are constant on conjugacy classes; in case $G = \mathbf{Z}/m\mathbf{Z}$ is a *cyclic* group, this point of view leads to the *discrete Fourier transform*.

  For any G, we'll describe how characters lead to interesting numerical information about G. And we'll point out some applications to statistical questions in case G is the symmetric group S_N.

- We consider **error-correcting codes** which we take to be vector subspaces C of the vector space ${\mathbf{F}_p}^N$, where $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ denotes the *finite field* with p-elements for a prime number p.

  An important parameter $d = d(C)$ is the *minimum (Hamming) distance* between two distinct codewords $v, w \in C$. Codes with larger values of d allow for more effective correction of *transmission errors*. We will give various algebraic constructions of codes – such as the *Reed-Solomon codes* – and describe their parameters.

- **formalization of mathematics.** Proof assistants – for example, *Lean* and *Agda* – provide a means of expressing and checking mathematical proofs via computer. For example, using tools like these a machine-checked proof of the *odd-order theorem* in group theory has been given.

  I plan to (i) have students learn to give some simple formalized proofs proofs in *Lean*, and (ii) explain enough *type theory* to describe (at least roughly) *how* a computer can represent and check proofs.

WHY WOULD YOU WANT TO TAKE THIS COURSE?

The course is aimed at mathematically inclined students with a good background in linear algebra and perhaps some familiarity with constructions in algebra. If you'd like to see examples of the utility of algebra in solving some interesting problems, the course is for you!