Section 1

Administration
Important Details

1. Klim Efremenko 37/303, email klim at bgu.ac.il
   Reception: Tue 12:00-14:00 (please coordinate via email in advance)

Course website: https://www.cs.bgu.ac.il/~klim/Teaching/Webpage
Make sure to read all what is there.
Important Details

1. Klim Efremenko 37/303, email klim at bgu.ac.il Reception: Tue 12:00-14:00 (please coordinate via email in advance)

2. Course website:
   https://www.cs.bgu.ac.il/~klim/Teaching/Webpage
   Make sure to read all what is there.
Goals of the Course


2. Team Work.

3. Training to understand the problem.

4. Picking the right working environment.

5. Asking the right questions.
Goals of the Course

2. Team Work.
Goals of the Course

2. Team Work.
3. Training to understand the problem.
Goals of the Course

2. Team Work.
3. Training to understand the problem.
4. Picking the right working environment.
Goals of the Course

2. Team Work.
3. Training to understand the problem.
4. Picking the right working environment.
5. Asking the right questions
How it works

1. During first 4 weeks you need to find your group and pick your project.
How it works

1. During first 4 weeks you need to find your group and pick your project.
2. The project could be done in groups of one two or three students.
How it works

1. During first 4 weeks you need to find your group and pick your project.
2. The project could be done in groups of one two or three students.
3. In at most 4 week need to pick a project.
How it works

1. During first 4 weeks you need to find your group and pick your project.
2. The project could be done in groups of one two or three students.
3. In at most 4 week need to pick a project.
4. After this each team should work on his topic and come to reception hours if need help.
How it works

1. During first 4 weeks you need to find your group and pick your project.
2. The project could be done in groups of one two or three students.
3. In at most 4 week need to pick a project.
4. After this each team should work on his topic and come to reception hours if need help.
5. By 01.08.2021 each group should submit the report on the project.
1. The grade will be composed from: 70% evaluation of the final project, and 30% evaluation of the progress of the pair throughout the semester.
Submition

1. Make sure to manage your time and submit report on time.
Submission

1. Make sure to manage your time and submit report on time.
2. Final submission should include all code you wrote and the report.
1. Make sure to manage your time and submit report on time.
2. Final submission should include all code you wrote and the report.
3. Projects’ reports should be written in English.
Submition

1. Make sure to manage your time and submit report on time.
2. Final submission should include all code you wrote and the report.
3. Projects’ reports should be written in English.
4. The report should include names and i.d. numbers.
1. Make sure to manage your time and submit report on time.
2. Final submission should include all code you wrote and the report.
3. Projects’ reports should be written in English.
4. The report should include names and i.d. numbers
5. The first part of the report should include problem definition, goals of the project. If applicable please explain what are other possible solutions to this problem and compare them.
Submition

1. Make sure to manage your time and submit report on time.
2. Final submission should include all code you wrote and the report.
3. Projects’ reports should be written in English.
4. The report should include names and i.d. numbers.
5. The first part of the report should include problem definition, goals of the project. If applicable please explain what are other possible solutions to this problem and compare them.
6. Second part should include explanation of how you solved the problem.
Submition

1. Make sure to manage your time and submit report on time.
2. Final submission should include all code your wrote and the report.
3. Projects’ reports should be written in English.
4. The report should include names and i.d. numbers.
5. The first part of the report should include problem definition, goals of the project. If applicable please explain what are other possible solutions to this problem and compare them.
6. Second part should include explanation of how you solved the problem.
7. The third part should include experiments. (. I encourage you to run experiments also in the range of parameters where there is no promise that algorithms will work.)
1. Make sure to manage your time and submit report on time.
2. Final submission should include all code your wrote and the report.
3. Projects’ reports should be written in English.
4. The report should include names and i.d. numbers
5. The first part of the report should include problem definition, goals of the project. If applicable please explain what are other possible solutions to this problem and compare them.
6. Second part should include explanation of how you solved the problem.
7. The third part should include experiments.(I encourage you to run experiments also in the range of parameters where there is no promise that algorithms will work.)
8. The last part of the report should include conclusions, and ideas for improvement.
Integrity

1. Few groups can work on the same project.

You can talk with each other in case you do so you need to specify it in your final submission; this has no affect on your grade.

You can not copy code of the other group.

You can not copy experiments of the other group.
Integrity

1. Few groups can work on the same project.
2. You can talk with each other in case you do so you need to specify it in your final submission this has no affect on your grade.
Integrity

1. Few groups can work on the same project.
2. You can talk with each other in case you do so you need to specify it in your final submission this has no affect on your grade.
3. You can not copy code of the other group.
Integrity

1. Few groups can work on the same project.
2. You can talk with each other in case you do so you need to specify it in your final submission this has no affect on your grade.
3. You can not copy code of the other group.
4. You can not copy experiments of the other group.
Section 2

Error Correcting Codes
What is Error Correcting Codes the basic motivation

- Alice wants to send a message to Bob.
- There is noise during the communication.
- Alice adds extra information so that Bob can decode despite the noise.
- Example: Alice repeats everything she says three times.
- Error Correcting Code is a way to encode information such that it will be resilient to noise.
What is Error Correcting Codes the basic motivation

- Alice wants send message to Bob.
- There is a noise during the communication.
What is Error Correcting Codes the basic motivation

- Alice wants to send a message to Bob.
- There is a noise during the communication.
- Alice adds extra information so that Bob can decode despite the noise.
What is Error Correcting Codes the basic motivation

- Alice wants to send a message to Bob.
- There is a noise during the communication.
- Alice adds extra information so that Bob can decode despite of noise.
- Example: Alice repeats everything she says three times.
What is Error Correcting Codes the basic motivation

- Alice wants to send a message to Bob.
- There is noise during the communication.
- Alice adds extra information so that Bob can decode despite the noise.
- Example: Alice repeats everything she says three times.
- Error Correcting Code is a way to encode information such that it will be resilient to noise.
What is Error Correcting Codes Applications

- Communication
What is Error Correcting Codes Applications

- Communication
- Storage
What is Error Correcting Codes Applications

- Communication
- Storage
- Crypto
What is Error Correcting Codes Applications

- Communication
- Storage
- Crypto
- Pseudo-Randomness
Section 3

Basic Definitions
Basic Definitions

- Code is a mapping $C : \Sigma^k \rightarrow \Sigma^n$. 

- Rate of the code $R = \frac{k}{n}$. How fast is your code.

- Example Repition code (repeat each symbol 3 times) has rate $\frac{1}{3}$.

- Linear Code if $\Sigma$ is a finite field and $C$ is a linear mapping. Almost all codes are linear.
Basic Definitions

- Code is a mapping $C: \Sigma^k \rightarrow \Sigma^n$.
- Rate of the code $R = \frac{k}{n}$. How fast is your code.
Basic Definitions

- Code is a mapping $C : \Sigma^k \rightarrow \Sigma^n$.
- Rate of the code $R = \frac{k}{n}$. How fast is your code.
- Example Repition code (repeat each symbol 3 times) has rate $\frac{1}{3}$. 

Klim Efremenko (BGU)
Basic Definitions

- Code is a mapping \( C : \Sigma^k \rightarrow \Sigma^n \).
- Rate of the code \( R = \frac{k}{n} \). How fast is your code.
- Example Repition code (repeat each symbol 3 times) has rate \( \frac{1}{3} \).
- Linear Code if \( \Sigma \) is a finite field and \( C \) is a linear mapping. Almost all codes are linear.
Basic Definitions

▷ Hamming Distance for $v_1, v_2 \in \Sigma^n$, $d_H(v_1, v_2) = \{i; v_1[i] \neq v_2[i]\}$. 

Distance of the code $d(C) = \min\{d_H(x, C(y)) : x \neq y\}$. 

Code with distance $d$ can detect $d-1$ errors and can correct $d-1/2$ errors. 

We want both $R$ and $d$ to be large usually this is contradicting goals.
Basic Definitions

- Hamming Distance for $v_1, v_2 \in \Sigma^n$, $d_H(v_1, v_2) = \{ i; v_1[i] \neq v_2[i] \}$.
- Distance of the code $d(C) = \min_{x \neq y} \{ d_H(C(x), C(y)) \}$.
Basic Definitions

- Hamming Distance for $\mathbf{v}_1, \mathbf{v}_2 \in \Sigma^n$, $d_H(\mathbf{v}_1, \mathbf{v}_2) = \{i; \mathbf{v}_1[i] \neq \mathbf{v}_2[i]\}$.
- Distance of the code $d(C) = \min_{x \neq y} \{d_H(C(x), C(y))\}$.
- Code with distance $d$ can detect $d - 1$ errors and can correct $\frac{d-1}{2}$ errors.
Basic Definitions

- Hamming Distance for $v_1, v_2 \in \Sigma^n$, $d_H(v_1, v_2) = \{i; v_1[i] \neq v_2[i]\}$.
- Distance of the code $d(C) = \min_{x \neq y}\{d_H(C(x), C(y))\}$.
- Code with distance $d$ can detect $d - 1$ errors and can correct $\frac{d-1}{2}$ errors.
- We want both $R$ and $d$ to be large usually this is contradicting goals.
Section 4

Selected Projects
List Decoding of Reed-Solomon

This project is recommended if it is your first research project.

- Reed Solomon is the most faumost Error-Correcting Code
List Decoding of Reed-Solomon

This project is recommended if it is your first research project.

- Reed Solomon is the most faumost Error-Correcting Code
- It is based on polynomials.
List Decoding of Reed-Solomon

This project is recommended if it is your first research project.

- Reed Solomon is the most faumost Error-Correcting Code
- It is based on polynomials.
- List decoding is a relaxition of decoding when instead of the one answer you output the list.
List Decoding of Reed-Solomon

This project is recommended if it is your first research project.

- Reed Solomon is the most faumost Error-Correcting Code
- It is based on polynomials.
- List decoding is a relaxition of decoding when instead of the one answer you output the list.
- You can achieve better rates than in case of unique decoding.