ניסוי ECC 79 - דוח הכנה חלק ב'

מגישים:
כריסטיאן שקור, 208157826, christian.s@campus.technion.ac.il
לארין עטאללה, 208653543, lareine.at@campus.technion.ac.il

שאלה ראשונה:

- **containers.Map**: אובייקט מסוג מפה הוא מבנה נתונים עם התאמה חח"ע בין מפתח לבין ערך. מאתחלים את המבנה ע"י קבלת וקטור המפתחות וקטור הערכים בעזרת הפקודה:(containers.Map(Keys, Values. ניתן לגשת לאיבר ספציפי ע"י גישה למבנה הנתונים במקום של המפתח הרצוי בעזרת mapName(key).
- **comm.BCHDecoder**: זוהי פונקציה אשר מייצרת מקודד מסוג BCH ומקבלת: אורך מילת קוד (n) ומימד קוד (k).
- **comm.RSDecoder**: זוהי פונקציה המייצרת מפענח מסוג (Reed-Solomon (RS או משנה ערכים באופן ישיר (ע"י ציון שם הפרמטר ומה הערך המצוין לו) ומקבלת: אורך מילת קוד (n), מימד הקוד (k), וקטור של מקדמי הפולינום בו משתמשים בקוד (GP) ואורך המילה המקוצרת לאחר הקידוד (s).
- **comm.RSEncoder**: זו פונקציה שמייצרת מקודד מסוג RS ומקבלת אורך מילת קוד (n), מימד הקוד (k), וקטור של מקדמי הפולינום בו משתמשים בקוד (GP) ואורך המילה המקוצרת לאחר הקידוד (s).
- **comm.BCHEncoder**: זוהי פונקציה המייצרת מקודד BCH עם מימד מסוים ואורך קוד מסוים. מקבלת: אורך מילת קוד (n) ומימד קוד (k).
- **ErasuresInputPort**: דגל שמסמן אפשרות של מחיקת חלקים מהקוד. אם הוא דלוק זה מאפשר לקבל כקלט וקטור שיסמן אילו ביטים בקוד למחוק.
- **BitInput**: דגל שמסמן אם הקלט מתקבל כמספר בינארי או כמספר שלם (int).
- **Histogram**: זו פונקציה שמייצרת היסטוגרמה עבור וקטור נתונים מסוים. הפונקציה מקבלת כקלט: וקטור הנתונים (X) ומספר העמודות (Nbins).

שאלה שניה:

- פרץ שגיאות הוא רצף של שגיאות (או מחיקות) במילות הקוד המשודרות, כך שיכולות להיות אף מילים שלמות שישתבשו.
- פעולת השזירה משנה את תזמון השידורים של מילים בקוד כך שגם במקרים בהם יתרחשו פרצי שגיאות ארוכים לא ימחקו מילים שלמות. פעולה זו למעשה מערבבת בין החלקים של המילים השונות בזמן השידור, ובכך מורידה את הסיכוי לפגיעה משמעותית בשידור.
- המחרוזת לפני השזירה: MatlabCodingFun
  המחרוזת לאחר השזירה: MlCiFaaonutbdgn
  המחרוזת לאחר דה-שזירה: MatlabCodingFun

```
str= 'MatlabCodingFun';
shzira = reshape(str, 3, 5);
shzira = shzira';
nStr = reshape(shzira, 1, 15);
mStr = reshape(nStr, 5, 3);
lStr = reshape((mStr)', 1, 15);
```

שאלה שלישית:

שאלה רביעית:

חסם סנגלטון הוא קשר שמקשר בין אורך הקוד n, כמות האינפורמציה המועברת בתוכו k ומרחק הקוד d. והוא הקשר הבא: d <= n - k + 1

מסקנה: הגדלת d ← הגדלת n או ← הקטנת k. ולהפך.

שאלה חמישית:

בקוד RS מגיעים ל- d = n - k + 1.

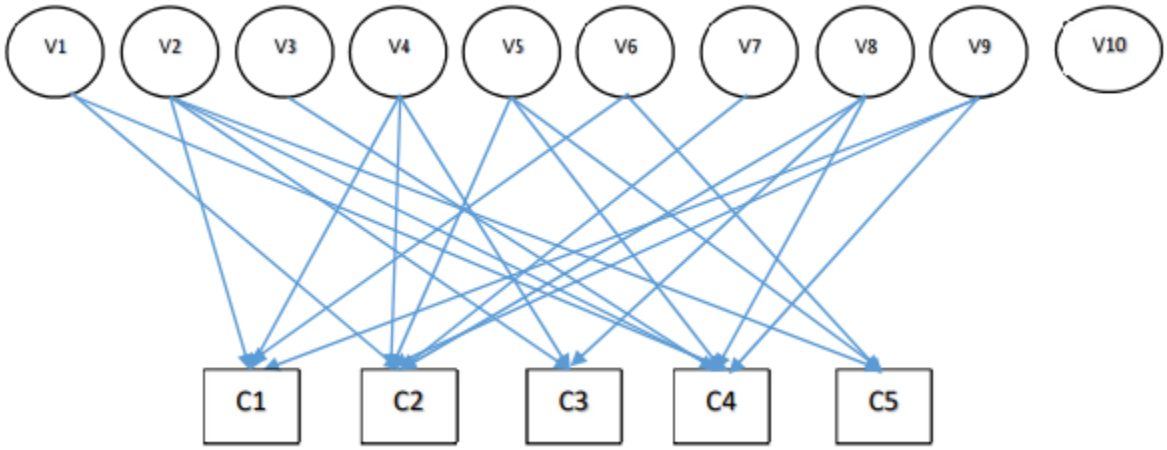
לפי משפט, אפשר לתקן d-1 שגיאות. מכאן, אפשר לתקן n-k שגיאות.

שאלה שישית:

- משתנה מסוג sparse הוא מטריצה דלילה. מטריצה דלילה היא מטריצה שמספר האפסים בה משמעותי. משתנה זה שומר רק את המקומות במטריצה שאינם אפסים ובכך חוסך בזיכרון. שימוש בו חסכוני כאשר רוצים לבצע איטרציות רבות על מטריצה כזו. בהקשר שלנו, עבור קודים משתמשים במטריצות דלילות משתמשים במשתנה זה כדי לקבל פענוח לינארי בגודל הקלט עם ביצועים טובים.

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	1	0	0	1	0
2	1	0	0	1	1	0	1	1	1	0
3	0	1	0	1	0	0	0	1	0	0
4	1	1	1	0	1	0	0	1	1	0
5	0	1	0	0	1	1	0	0	0	0

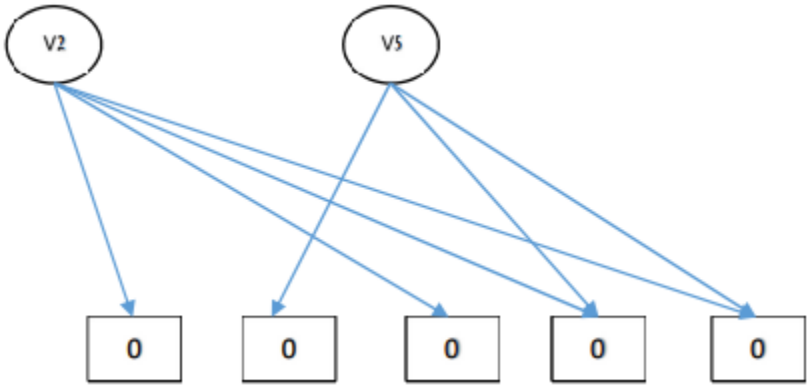
- גרף הטאנר המתאים לה:



וקטור האפס תמיד יהיה מילה חוקית בקוד מכיוון שהוא תמיד מקיים את המשוואה:

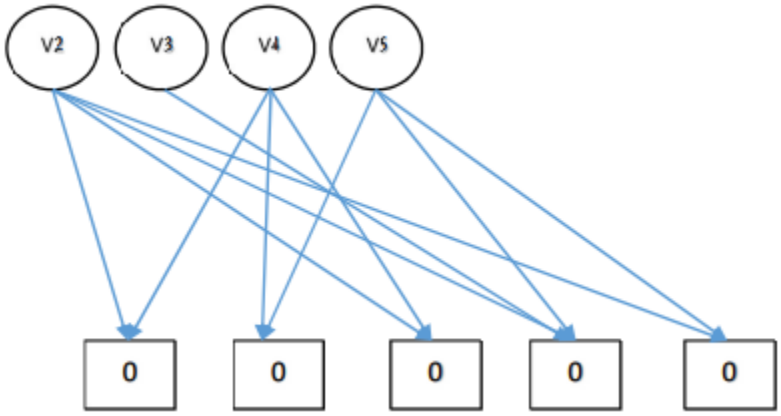
Hc^(Transpose) = 0

- נתחיל מאיפוס כל ערכי האליפסות, ונשים לב כי כל צמתי המשתנה הידועים הם 0 ,לכן פעולת ה-xor תשאיר את צמתי הבדיקה מאופסים גם. נקבל את המצב:



נתחיל מצומת הבדיקה c1 ששולח 0 ל-v2 ,ולכן מתקבל: v2=0 .בצורה דומה c2 שולח 0 ל-v5 ולכן v5=0 .לכן סה"כ נקבל שמילת הקוד החסרה היא וקטור האפס.

- בצורה דומה, נקבל את המצב:



במצב הנתון אין שום צומת בדיקה מדרגה 1 ,לכן האלגוריתם נתקע ולא נוכל לקבל מילת קוד