מערכת מבוססת עץ החלטה לזיהוי ספרות – סקירה כללית

בגרסת המימוש השנייה, בחרנו לבדוק עבור כל פיקסל נתון, בנוסף לערכו, גם את ערכם של הפיקסלים הסמוכים לו.

לדוג', עבור פיקסל 324, נרצה לחשב בנוסף לערכו גם את ערכם של הפיקסלים הבאים:

295	296	297
323	324	325
351	352	353

לצורך כך, נדרשנו לחשב את העמודה והשורה שמס' פיקסל נתון מייצג, ולחשב את מספרי הפיקסלים של העמודות והשורות הסמוכים.

כמובן שנדרשנו לחישוב ספציפי עבור מקרים שונים בהם פיקסל נתון הוא בשולי הדף.

לדוג' עבור פיקסל 5, קיימים 5 תאים סמוכים בלבד, שכן לא קיימת שורה מעליו:

4	5	6
32	33	34

עניינו של השינוי אל מול האלגוריתם הראשון, מתבטא בכך שהפעם התנאי אינו מוגדר כבדיקה האם ערכו של פיקסל נתון גדול מ – 128.

אלא, האם ממוצע ערכי הפיקסלים הסמוכים לפיקסל נתון גדול מ – 128.

טרם ההחלטה להשתמש באפשרות המצויינת לעיל בגרסת המימוש, ביצענו ניסיונות שונים ומגוונים, להלן חלקם העיקרי:

(1) **<u>סיווג הפיקסלים השונים בהקשר ההשפעה שלהם על התוצאה:</u>**

פיקסלים שמייצגים מידע ב"קצוות התמונה" למשל, משמעותיים הרבה פחות אל מול פיקסלים שנמצאים במרכז התמונה.

ראינו כי אופן מימוש זה לא משפר ואף מעלה את אחוז השגיאה, לצד עלייה ניכרת בזמן הריצה.

(2) <u>מציאת מכנה משותף בין רשומות עם תוויות זהות:</u>

ניסינו לאפיין האם קיים מכנה משותף כלשהו המייחד כל אחת מהספרות, ולהטמיע את התובנות שלנו כתנאים. לאחר ניסיונות שונים ומגוונים, הבנו כי ללא הנחות נוספות על הקלט (מיקום הספרה ביחס לדף שעליו היא מופיעה למשל, או גודל הספרה ביחס לדף שעליו נכתבה), יהיה לנו קשה עד בלתי אפשרי ליצור עץ החלטה שיהיה רלוונטי לכל קלט.

(3) **<u>שינוי גודלו של P</u>**:

בחנו את תפקודם של ערכי P שונים. ערכו של P משפיע על גודל העץ הנבחר. בסופו של דבר, עבור מדגם אימון גדול מספיק, נראה היה שגודל העץ נשאר קבוע ולא משתנה.

(4) **<u>שינוי הערך "128" :</u>**

ניסינו לבחון את התנהגות האלגוריתם עבור ערכים שונים, בעיקר נמוכים יותר מ – 128.

(5) **<u>גרסאות שונות של השיטה הנבחרת :</u>**

החלטנו לשמר את הרעיון הכללי של בחינת הפיקסלים שנמצאים "בסביבת" פיקסל נתון. טרם ההחלטה על הגגרסא הספציפית שנבחרה, ניסינו לבחון גם גרסאות שונות:

א. "ספירת" הפיקסלים שערכם גדול מ – 128 ובדיקה האם הם מהווים את רוב הפיקסלים מסביב לפיקסל נתון (במקום חישוב הממוצע).
ב. בעת בדיקת פיקסל נתון, התייחסות לסביבת הפיקסל בלבד, תוך השמטת הפיקסל הנתון.

במהלך כתיבת הפרויקט, נתקלנו באתגרים שונים. אלו האתגרים העיקריים:

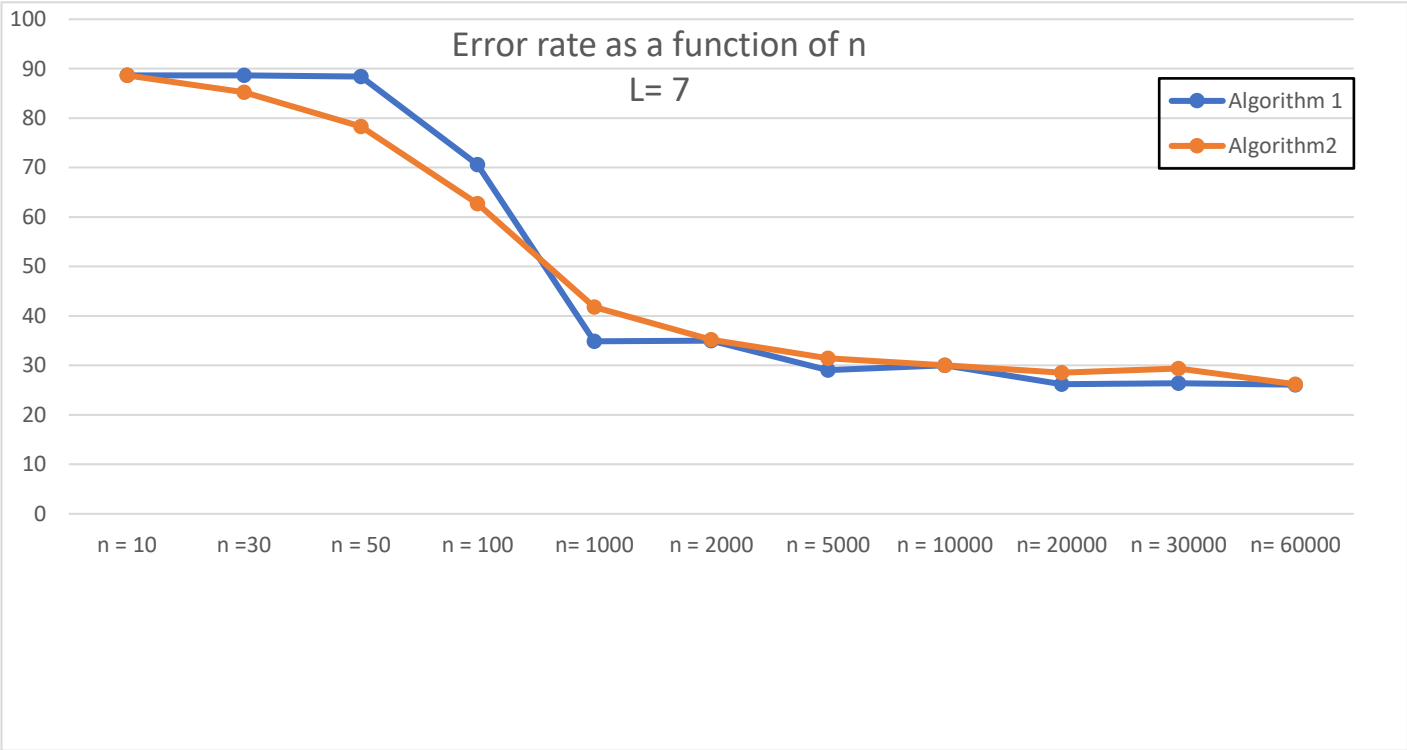
<u>זמן ריצה –</u>

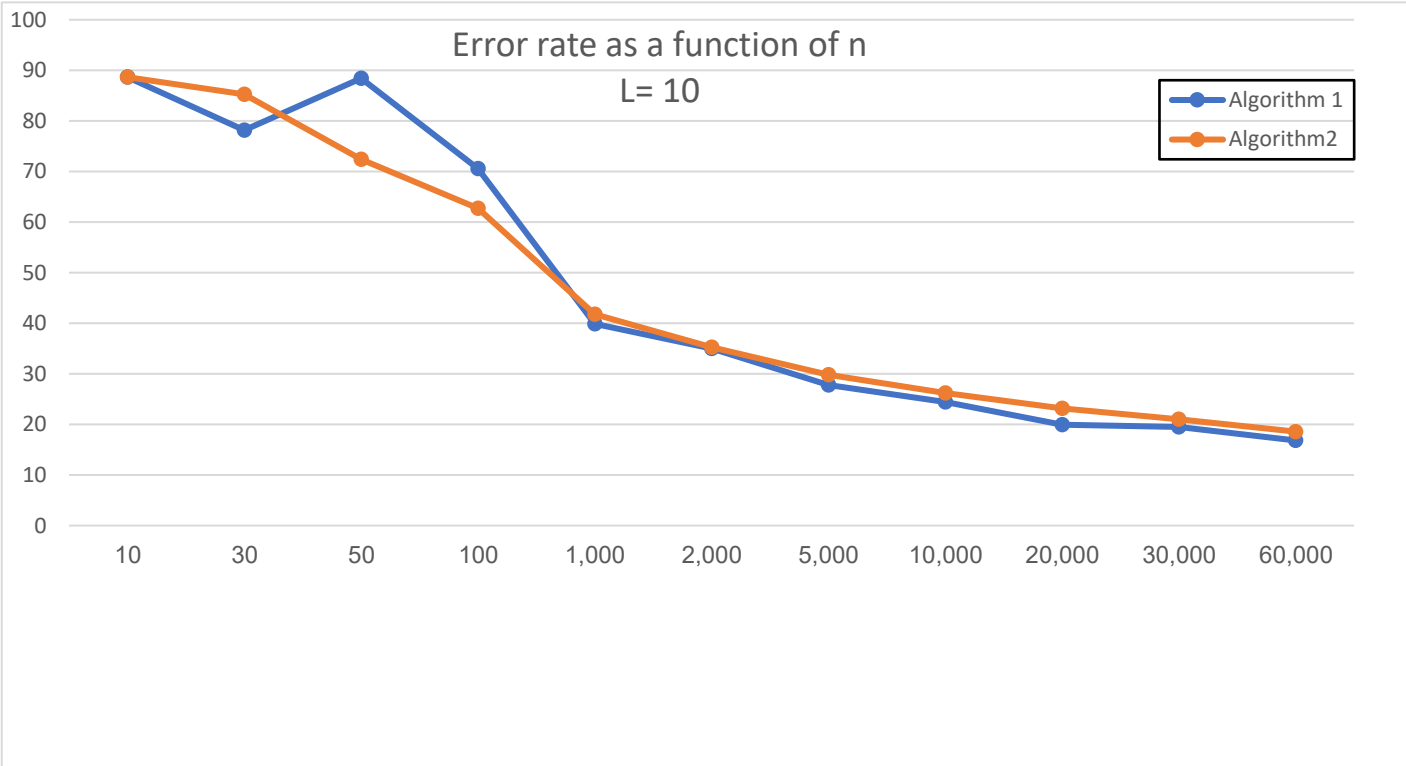
לאור העובדה שהאלגוריתם נדרש לעבד כמות גדולה יחסית של מידע (כ – 60,000 רשומות המכילות כל אחת 784 אינדקסים), כל פעולה שבתכנות רגיל לא מקבלת "טיפול מיוחד" בהקשר זמני ריצה כמו מעבר על כלל הפיקסלים של תמונה נתונה, הופכת למשמעותית.

הבנו שהמפתח לקיצור זמן הריצה הינו ע"י ביצוע Trade-Off אל מול משאבי זיכרון.

לכן, בחרנו להשתמש בכמות גדולה יחסית של זיכרון ע"מ להרוויח זמני ריצה יעילים יותר בפעולות שונות.

היישום של עיקרון זה בא לידי ביטוי, למשל, בהגדרת המחלקה "Node" – בשונה מעץ בינארי רגיל, הוספנו למחלקה שדות נוספים שישמרו מידע שכבר חושב בעבר, למשל רשימת האינדקסים של רשומות שכבר הגיעו לאותו Node ספציפי.


Error rate as a function of n
L= 7
Algorithm 1
Algorithm2
100
90
80
70
60
50
40
30
20
10
0
n = 10
n =30
n = 50
n = 100
n= 1000
n = 2000
n = 5000
n = 10000
n= 20000
n = 30000
n= 60000


Error rate as a function of n
L= 10
Algorithm 1
Algorithm2
100
90
80
70
60
50
40
30
20
10
0
10
30
50
100
1,000
2,000
5,000
10,000
20,000
30,000
60,000

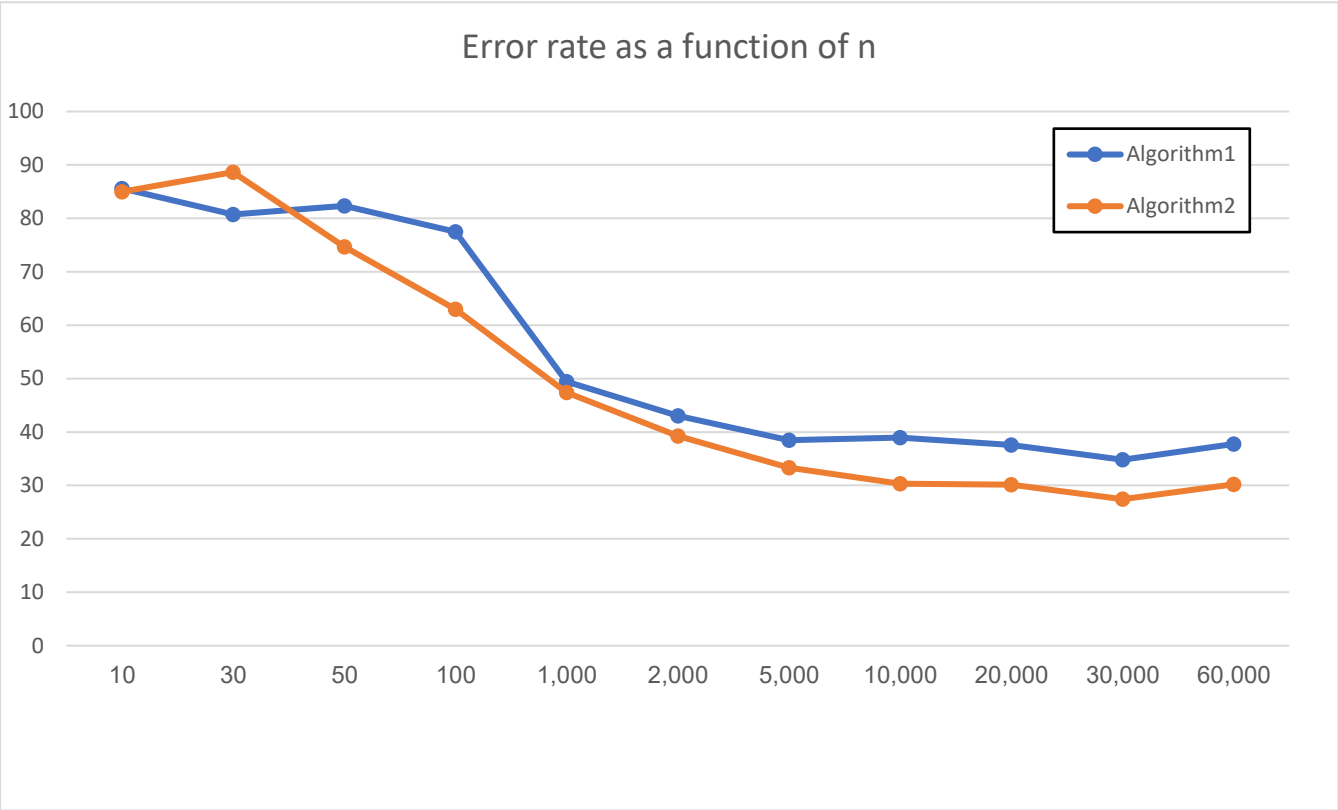
גירסה ראשונה : גודל המדגם = 10,000, P =10%, L =7

	0	1	2	3	4	5	6	7	8	9
0	791	0	25	11	5	47	59	21	9	12
1	0	1057	10	2	1	9	9	11	36	0
2	0	25	618	19	24	17	78	38	145	34
3	20	16	43	598	5	185	20	29	32	62
4	5	5	13	35	701	12	61	30	14	106
5	39	5	24	225	36	452	51	19	14	27
6	27	10	51	8	50	28	748	0	23	13
7	5	11	49	3	29	18	8	813	19	73
8	27	23	38	29	44	29	167	25	538	54
9	8	8	8	45	47	51	18	124	21	679

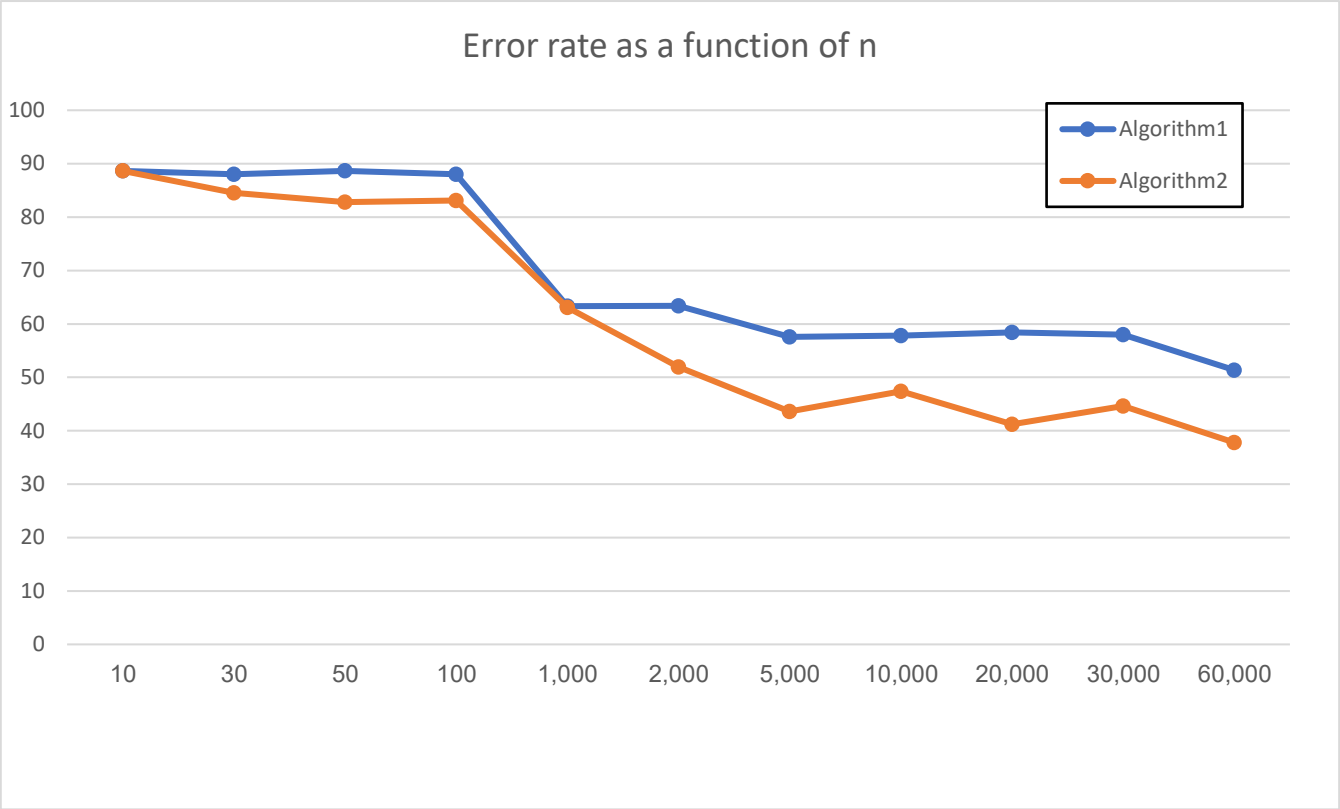
גירסה שנייה : גודל המדגם = 10,000, P =10%, L = 7

	0	1	2	3	4	5	6	7	8	9
0	820	0	10	23	16	27	31	11	7	35
1	0	1037	33	18	0	8	5	10	24	0
2	30	20	711	34	18	25	62	58	52	22
3	22	5	20	714	4	130	9	20	65	21
4	7	3	19	34	673	37	45	13	20	131
5	42	5	21	113	36	523	50	12	42	48
6	48	9	40	7	37	31	708	16	53	9
7	1	6	42	26	14	12	4	843	12	68
8	9	17	57	47	38	65	54	21	592	74
9	10	5	9	48	61	47	14	32	25	758

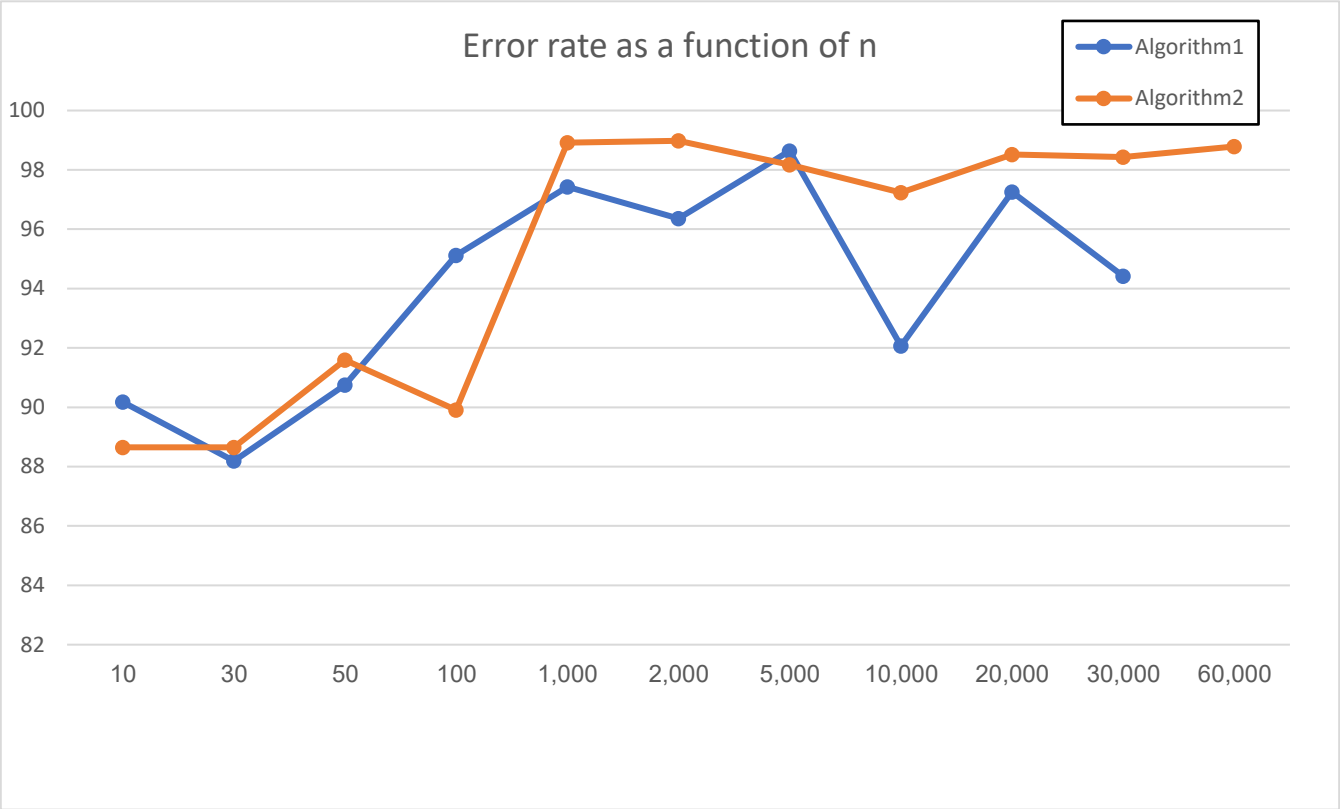
$: \alpha = 0.1$


Error rate as a function of n
100
90
80
70
60
50
40
30
20
10
0
10
30
50
100
1,000
2,000
5,000
10,000
20,000
30,000
60,000
Algorithm1
Algorithm2

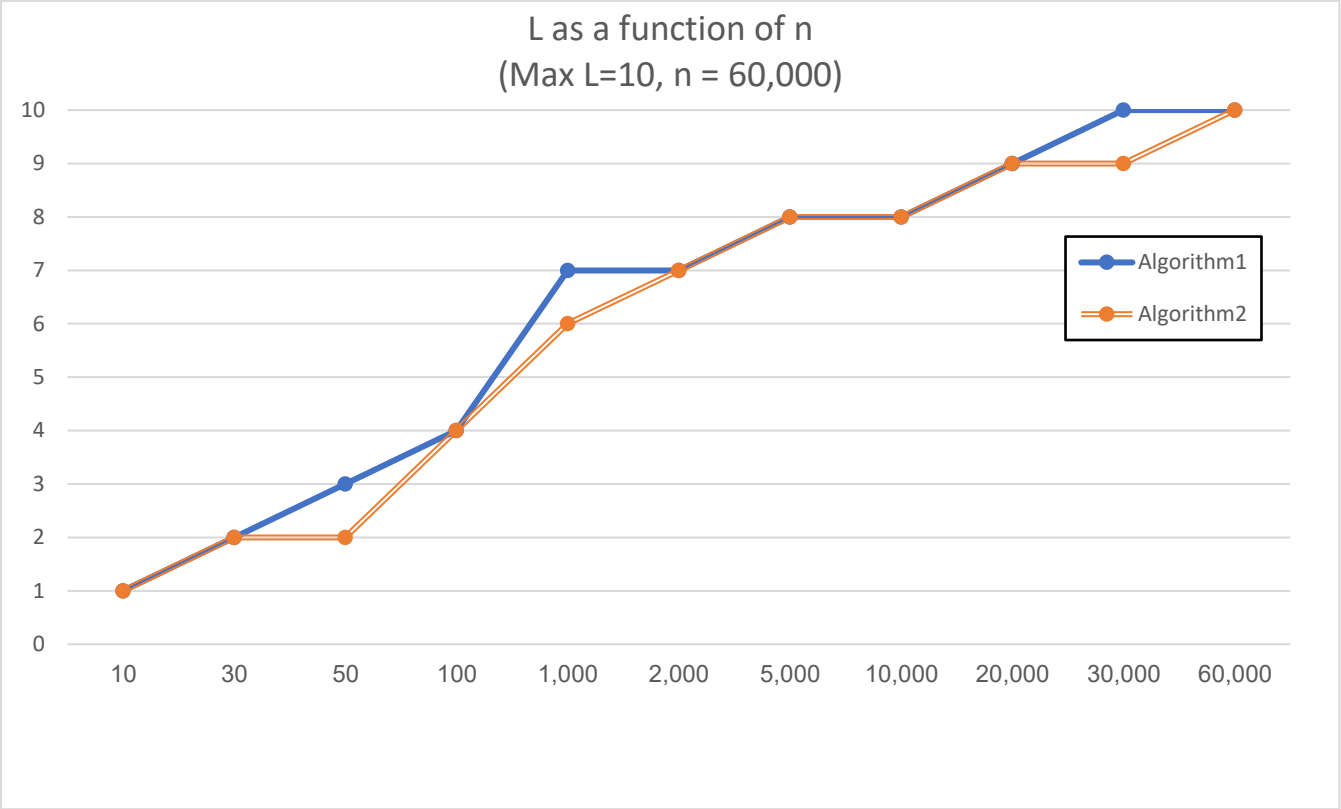
$: \alpha = 0.3$


Error rate as a function of n
100
90
80
70
60
50
40
30
20
10
0
10
30
50
100
1,000
2,000
5,000
10,000
20,000
30,000
60,000
Algorithm1
Algorithm2

: $\alpha = 0.6$


Error rate as a function of n
Algorithm1
Algorithm2
100
98
96
94
92
90
88
86
84
82
10
30
50
100
1,000
2,000
5,000
10,000
20,000
30,000
60,000

גודל העץ כתלות בגודל המדגם:



גודל העץ כתלות במקדם הרעש :