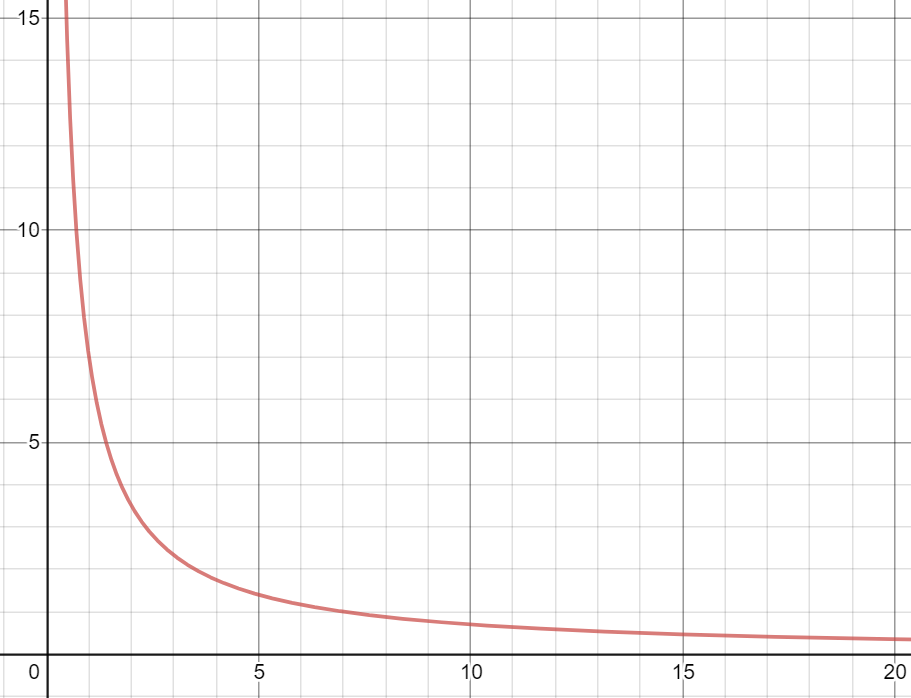
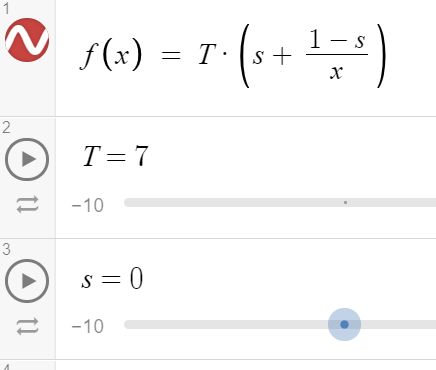
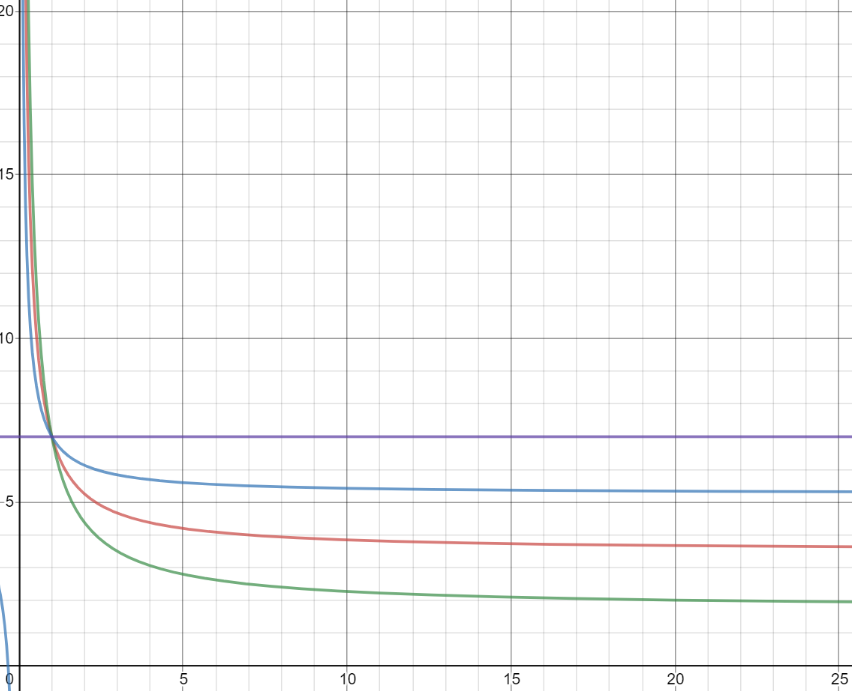
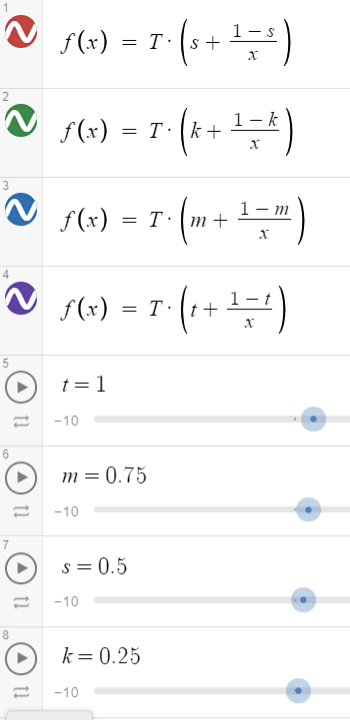
3.2.1)According to Amdahl’s law,

Latency of A in Y axis, affected by the number of threads in X axis,  
under the assumption A is fully parallelable, meaning s=0: (T1 is arbitrarily 7)



3.2.2)



3.2.3)

הראנו ניתוח עד הפעלה של 20 חוטים כי זה מספר השורות בלוח ועל כן זה המספר האפקטיבי המקסימאלי.  
עבור הלוח הקטן ניתן לראות שאפילו מעבר מחוט אחד לשניים פוגע בביצועים. ומעבר לכך ישנה מגמה מונוטונית עולה של זמן חישוב כולל לאור הוספת חוטים, למה?  
על אף שעברנו לחישוב מקבילי של שני חצאים של המשימה, הלוח עצמו די קטן, מכך יוצא שזמן יזימת חוט נוסף ואתחולו, גובר על הזמן שהיה לוקח לחוט אחד לבצע משימה מלאה (לעומת חצי משימה). כלומר – זמן החישוב לחצי לוח הוא קטן מזמן הקצאת עוד חוט (פעולת גרעין הכוללת הקצאת מחסנית וכו).  
תחת הנחה זו, כמובן שנראה עלייה נוספת בזמן הכולל ככל שנוסיף עוד חוטים. על כן, מספר החוטים האופטימאלי ללוח זה הוא 1.  
לגבי זמן ממוצע לחוט - כמובן שפוחת ככל שיש יותר חוטים. כל אחד בעולמו הקטן מחשב פחות שורות ולכן מתבצע בזמן קצר יותר.  
גם אם היינו מוסיפים מעבדים נוספים, לא היינו מקבלים שיפור בביצועים כי הבעיה לא במקבול החישוב ומשאבי עיבוד פנויים אלא בזמן אתחול חוט.

ג

הראנו ניתוח עד הפעלה של 64 חוטים כי זה מספר השורות בלוח ועל כן זה המספר האפקטיבי המקסימאלי.  
ראשית אתייחס לקפיצות הקטנות שחורגות מקו המגמה. בתרגיל זה, מדיניות החלוקה של השורות בין החוטים אומרת לחלק את השורות בכמות החוטים ואת השארית להעביר לחוט האחרון. לכן עבור 32 אנחנו רואים ירידה בזמן ביצוע כי יש חלוקה מושלמת של 2 שורות לחוט. עבור מספר כמו 22 ניתן לראות עלייה בזמן ביצוע בגלל שכל חוט מקבל 2 שורות, והחוט האחרון מקבל 20 (בכל דור יש המתנה לחוט זה שיסיים).

ניתן לראות שיפור בביצועים בהוספת החוטים הראשונים בזכות חלוקת העבודה בין מעבדים. מספר החוטים האופטימאלי ללוח זה הוא 4.  
הסבר אפשרי הוא שאחרי יותר מ-4 שורות אנחנו חוזרים לבעיה מהלוח הקטן בו עלות היזימה של עוד חוטים גדולה יותר מהזמן שהיה לוקח לפחות חוטים לבצע משימות גדולות יותר.  
מסומן על הגרף קו מגמה לינארי וברור כי הוא עולה מונוטונית והוספת עוד חוטים פוגעת בביצועים.  
לגבי זמן ממוצע לחוט, (כמו ביתר הלוחות) פוחת ככול שיש יותר חוטים כי המשימה שיש לכל חוט לבצע קטנה יותר.

כמו בלוח הבינוני, גם כאן ניתן לראות "מדרגות" בביצועים בגלל מדיניות החלוקה.  
בגלל שלוח זה גדול במיוחד, אנחנו רואים קורלציה בין הגדלת כמות החוטים לירידה בזמן הביצוע – כמגמה. הוספתי את 7 הריצות עם ממוצע זמן הריצה הטוב ביותר, וכמות החוטים בכל אחת, ניתן להסביר את הקפיצות בין הכמויות חוטים בפסגת הביצועים על ידי אותה מדיניות חלוקה, וגם על ידי קצת מזל, הרי המערכת המריצה את הקוד אינה מבודדת ומטפלת בעוד תהליכים ופסיקות ובכלל יכולה לסבול מהאטה מסיבות אחרות ולכן גם אותה כמות חוטים יכולה להפיק תוצאות שונות בניסויים חוזרים. (מצרף טבלה של ריצות שונות של הלוח הקטן עם 20 חוטים כהוכחה). לפי הניתוח המספר האופטימאלי (מבין הבדיקות) של חוטים הוא 100. כאן לא הגענו לחסם כמו בלוחות הקודמים כי הלוח ענק והזמן שלוקח לכל חוט לבצע משימה עדיין גדול מזמן יזימה של חוט נוסף על מנת לחלק את העבודה.  
הגרף האפור דומה לגרף מסעיף ב' עם מקדם סדרתי . (מקבילי מאוד אך לא לגמרי).  
בכל זאת הקוד אינו מקבילי לחלוטין – יש הקצאת משימות שמבוצעת על ידי חוט ראשי.

|  |  |  |
| --- | --- | --- |
| # | #thread | Latency A |
| 1 | 100 | 9162.33 |
| 2 | 62 | 9295.7 |
| 3 | 71 | 9310.04 |
| 4 | 66 | 9326.31 |
| 5 | 83 | 9379.89 |
| 6 | 37 | 9608.49 |
| 7 | 99 | 9612.55 |

|  |  |
| --- | --- |
| #threads | Latency A |
| 20 | 922.54 |
| 20 | 869.33 |
| 20 | 893.49 |
| 20 | 925.35 |
| 20 | 948.13 |
| 20 | 813.94 |