אלגוריתם 1:

פונקציה ראשית מקבלת שני קבצים ומספר טבעי:

קובץ בו נמצאים מאקים לחיפוש: "addressOfFile\_Where\_I\_need\_To\_Get\_MAC"

קובץ בו נתונים כל הסריקות: "addressOfFile\_To\_Filter\_With\_MAC"

כמות השורות לחישוב שרת: "howMuch\_Powerful\_Wifis\_To\_Use\_For\_Calculation"

שלבי חישוב:

כל קבצים יהפכו לARRAYLISTים בכדי קריאת הקבצים חד פעמית. כל מאק ניקח כפילטר איתו נרוץ בליסט סריקות ונקח נתונים לפי כמות השורות הרצויה והשוואה של סיגנלים. ליסט שיתקבל לאחר פילטור של הקובץ יעבור לחישוב בפונקציית עזר לפי אלגוריתם הנתון במטלה. נקודה מחושבת עוברת לליסט חדש שבסוף יעבור להדפסה לקובץ במיקום שקובץ פילטרים היה בו עם תוספת " \_Our\_Algo1.csv" לשם

אלגוריתם 2:

פונקציה ראשית מקבלת שני קבצים ומספר טבעי:

קובץ בו נמצאים מאקים לחיפוש: "addressOfFile\_Where\_I\_need\_To\_Get\_MAC\_and\_SIGNAL"

קובץ בו נתונים כל הסריקות: "addressOfFile\_To\_Filter\_With\_MAC\_and\_Signal\_I\_get"

כמות השורות לחישוב שרת: "howMuch\_Powerful\_Wifis\_To\_Use\_For\_Calculation"

שלבי חישוב:

כל קבצים יהפכו לARRAYLISTים בכדי קריאת הקבצים חד פעמית. קובץ של פילטרים יעבור פילטור נוסף של שורות שאותם מפרקים לליסט של מאקים וליסט של סיגנילים. לאחר פירוק של כל שורה עושים פילטר של ליסט סריקות לפי ID שנמצא באותה שורה של פילטרים. על ליסט הנוצר רצים כל פעם עם מאק חדש מתוך רשימה ומייצרים ליסט של שורות לפי התאמה של זמן. ליסט שנוצר מעבירים לפונקציה שמחשבת התאמה של כל שורה לפי ליסט של סיגנלים. לאחר מכן בוחרים שורות המתאימות ביותר לפי מספר שורות לחישוב. לאחר מכן פונקציה נוספת מחשבת מיקום של מכשיר לפי אלגוריתם הנתון במטלה 2.

קבועים:

לא עשינו שום שינוי בקבועים מאיך שהם היו מוגדרים באלגוריתם הנתון.

|  |  |
| --- | --- |
| Value | Parameters |
| 2 | power |
| 10000 | norm |
| 0,4 | sig diff |
| 3 | min diff |
| -120 | no signal |
| 100 | diff no sig |