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:

נכניס את נתוני הקבצים לרשימות מקושרות, רשימה אחת של של מאקים ורשימה שניה של נתונים שנרוץ עליהם.

לכל כתובת מאק נסרוק את רשימת הנתונים בכדי למצוא את הכמות הרצויה של סריקות עם הסיגנל הכי גבוה.

הרשימה שנקבל לאחר הסינון תעבור לחישוב בפונקציית עזר לפי אלגוריתם הנתון במטלה.

לאחר חישוב הנקודה היא תעבור לרשימה שמיועדת להדפסת הקובץ CSV של האלגוריתם שלנו, קובץ זה ישמר בתיקייה של קובץ המאקים 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 |