מבוא לתכנות מונחה עצמים – מטלות קורס

מסמך זה מפרט את מכלול המטלות של קורס מבוא לתכנות מונחה עצמים, הרעיון המרכזי במטלות הקורס שהן למעשה מטלה אחת "מתגלגלת" כך שאתם למעשה תתבקשו כל הזמן לשפר ולהרחיב את המטלות הקודמות שלכם כדי לאפשר למידה מעמיקה ומתמשכת.

הנחיות כלליות:

1. את המטלה עושים בזוגות, יש להגיש את כל המטלות בזמן! לפי הנחיות, על כל איחור לא מוצדק תהיה הורדת ניקוד.
2. המטלות תיבדקנה באמת במהלך התרגולים, על כל אחד מבני הזוג להבין באופן מלא ושותף אל כל רכיבי המטלה בפרט כיצד להריץ לבדוק ולהכיר כל שורה בקוד.
3. המטלות תיבדקנה באופן אוטומטי באספקטים של "העתקות קוד" אין לבצע שום העתקה של קודים בין קבוצות שונות, מותר לעשות שימוש בקוד פתוח, אבל חובה לציין זאת בפירוש ולהביא את המקור המדויק. למען הסר ספק: שימוש בקוד פתוח (או כל קוד זמין ברשת) שלא יצוין מקור הקוד יחשב כהעתקה!
4. כלל הפיתוח יעשה בכלי בקרת התצורה של github, הכירו היטב את הכלי ועשו בו שימוש משמעותי ומעמיק, הן לקוד והן לתיעוד מסודר של הפרויקט שלכם.

המטלה עצמה:

במטלה זאת נפתח (בהדרגה) מערכת מורכבת שמאפשרת איסוף מידע גיאוגרפי הפקה של תובנות ממידע זה והצגת המידע בכלים גרפיים.

נסתכל על אפליקציה כגון waze, היא מפיקה מידע לגבי עומסי התנועה בזמן אמת ע"י צבירת המידע מנהגים רבים, וטיוב שלו.

בדומה הסתכלו על האפליקציות הבאות:

OpenStreenMap, G-Mon, openSignal כולן מאפשרות איסוף מידע גיאוגרפי מגוון בשיטות בגישות שונות, המידע כולל: עוצמת קליטה של הטלפון, נתוני גלישה, מיקום, מהירות, מיפוי ומידע נוסף שהמשתמש מעלה. שימו לב שיש המון סוגים של "אפליקציות גיאוגרפיות" חלקן אוספות מידע על טיולים – נניח "עמוד ענן", אחרות על "איסוף מדדים גופניים, דופק, מהירות כו'", ואפילו אפליקציות לאיסוף מידע של מחשב הרכב כגון TORQUE

באופן כללי נוכל לחלק את האפליקציות הללו לשלושה חלקים:

- אפליקציית לקוח (לרוב אפליקציית אנדרואיד) שאוספת את המידע ומעלה אותו "לשרת".
- "שרת" שמאפשר שמירה של נתוני המשתמשים, טיוב ובעיבוד שלהם.
- מערכת "תצוגה" וניהול שכוללת ממשק גרפי – לרוב בממשק של אפליקציית רשת.

מטלה 2 – גרסה מעודכנת 13.12.2017:

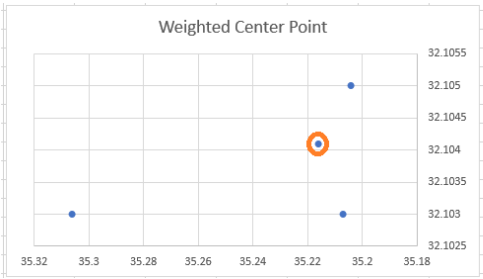
במטלה זאת נרחיב את מטלה 1 בתחום תכנון ומימוש אלגוריתמים, ע"י כך שנוסיף שני אלגוריתמים בסיסיים לעבודה עם נתונים גיאוגרפים: לשערוך המיקום של כל נתב וזיהוי מיקום לפי דגימות נתבים.

מומלץ לתכנן לפני שמתחילים לממש. חשוב להתחיל בהבנת מרחב הבעיה מומלץ לשחק בקובץ ה excel המצורף ולאחר שהבנתן לבצע תכנון בדמות כתיבת של מסמך קצר שבו תציינו איזה מחלקות וקבצים יהיו במערכת שלכם, מה כל רכיב יעשה, ומה יהיו הקשרים ביניהם. וכמובן פירוט לגבי אופן מימוש האלגוריתמים שלכם.

1. חלק ראשון נבצע דוח ביקורת על עבודות אחרות שנעשו: את הדוח יש לבצע באופן הבא: כל זוג יקבל שתי מטלות אחרות לבדיקה, ויעלה את חוות הדעת שלו לפי הסעיפים הבאים:
   - איכות העבודה מבחינת הפרויקט ב github
   - איכות התיעוד ומסמכי ההסבר.
   - איכות הקוד מבחינת חלוקה למחלקות והנדסת תוכנה.
   - איכות המערכת – יש להריץ ולבדוק את איכות התוצאות
   - הערות טקסט כלליות.

   את הדוח יש להעלות בטופס הבא:

2. חלק שני: מימוש אלגוריתמים: בחלק זה נממש שני אלגוריתמים על מידע גיאוגרפי:
   a. בהינתן מזהה WiFi (MAC) נשערך בעזרת המידע הקיים את המיקום שלו:
   - דרך נאיבית לעשות זאת היא לחפש באוסף המידע שלנו את כל הדגימות עם ה MAC המבוקש, ולהחזיר את המיקום עם הדגימה הכי חזקה.
   - דרך טובה יותר היא לחפש את מספר הדגימות (נניח עד 4) הכי חזקות של ה MAC המבוקש ולחשב ממוצע משוקלל בעזרתן – לפי הדוגמא הבאה:



	Lat	Lon	Alt	Sig
1	32.103	35.208	650	-50
2	32.105	35.205	660	-40
3	32.103	35.307	680	-90
w-center	32.10409	35.21701	658.667	

	wLat	wLom	wAlt	weight
1	0.012841	0.014083	0.26	0.0004
2	0.020066	0.022003	0.4125	0.000625
3	0.003963	0.004359	0.083951	0.000123
sum	0.03687	0.040445	0.756451	0.001148
w-sum	32.10409	35.21701	658.667	

נתייחס לקלט של שלוש דגימות כל אחת במיקום אחר ובעוצמה אחרת (של אותו MAC), ניתן לחשב את המיקום שלו לפי ממוצע משוקלל: כאשר המשקל של כל נקודה יהיה לפי אחד חלקי ריבוע העוצמה. שחקו עם קובץ ה excel וודאו שאתם מבנים היטב את הרעיון.

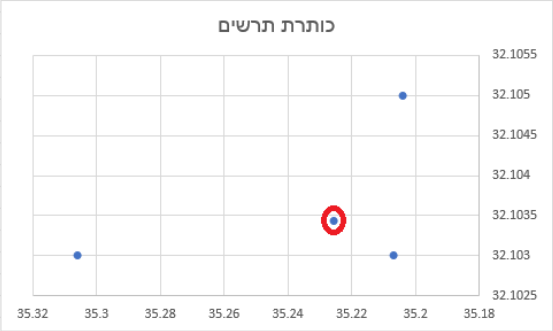
   b. בהינתן מספר דגימות של WiFi ועוצמת סיגנל (נניח 3 זוגות) נרצה לשערך את מיקום המשתמש. בסעיף זה נשתמש באותו אלג' פשוט של מרכז כובד משוקלל בשילוב עם שיטה לבדיקת מידת ההתאמה של כל אחת מהדגימות לקלט שלנו, להלן דוגמא קונספטואלית (דוגמא מלאה נמצא בקובץ ה excel המצורף)

	M1	Sig 1	M2	Sig2	M3	Sig3	W
Input	1	-50	2	-60	3	-80	
Data	1	-40	2	NONE	3	-60	
		(50-10)/50		0.1		(80-20)/80	0.8*0.1*0.75

בדרך זאת ניתן לדרג כל אחת מהדגימות ביחס למידת הדמיון לקלט שלנו, ואז ניקח את 4 הדגימות הכי דומות ונחשב את מרכז הכובד המשוקלל בניהן לפי הנוסחה של סעיף א'.

Parameter	Value
power	2
norm	10000
sig diff	0.4
min diff	3
no signal	-120
diff no sig	100

	sig1	diff1	w1	sig2	diff2	w2	sig3	diff3	w3	pi
Input	-50			-70			-90			
Data1	-62	12	1.480429	-79	9	0.847436	-71	19	0.380201	0.476989
Data2	-82	32	1	-120	100	0.323448	-82	8	0.537377	0.173813
Data3	-120	100	0.633957	-89	19	0.628496	-73	17	0.397498	0.158379
Data4	-91	41	0.905621	-120	100	0.323448	-120	100	0.195666	0.057315
Data5	-120	100	0.633957	-120	100	0.323448	-120	100	0.195666	0.040122



	Lat	Lon	Alt
1	32.103	35.208	650
2	32.105	35.205	660
3	32.103	35.307	680
	32.10343	35.22673	658.0198

	wLat	wLom	wAlt	weight
1	15.31276	16.79381	310.0426	0.476989
2	5.580275	6.119096	114.7168	0.173813
3	5.084444	5.591891	107.6978	0.158379
sum	25.97748	28.5048	532.4571	0.809181
w-sum	32.10343	35.22673	658.0198	

דוגמא לשיטה לשערוך מיקום מקלט WiFi בעזרת דירוג הדגימות ובחירת שלוש הדגימות הדומות ביותר וביצעו ממוצע משוקלל בניהן. שיטה זאת מבוססת על מספר פרמטרים – בורוד למעלה. באופן כללי ניתן לראות שרק שלוש הדיגמות העליונות הן הדומות ביותר לדגימת הקלט שלנו לפיכך השתמשנו בהן כדי לשערך את מיקום המקלט.

3. חלק שלישי: הרצה בדיקות וכתיבת דוח ניסוי השוואתי

בסעיף זה עליכם לבדוק את שני האלגוריתמים שלכם ולכתוב דוח בנושא – הדוח צריך לכלול את הסעיפים הבאים:

3.1 הסבירו בקצרה את שני האלגוריתמים שמימשתם, כולל פירוט לגבי הקבועים המוגדרים באלגוריתמים – ציינו את הערכים האופייניים של כל פרמטר.

3.2 הורידו והריצו את הפתרון שניתן לכם: קובץ בתיקייה testing שמכילה שני קבצי קלט בשמות: _comb_all_BM2_ ,_comb_all_BM3_. ושני קבצי בדיקה בשמות comb_no_gps_ts1, ותיקיית פלט עם שישה קבצי פלט.
הריצו את הדוגמאות כפי שמתואר באיור מטה, לפי התבניות הבאה:

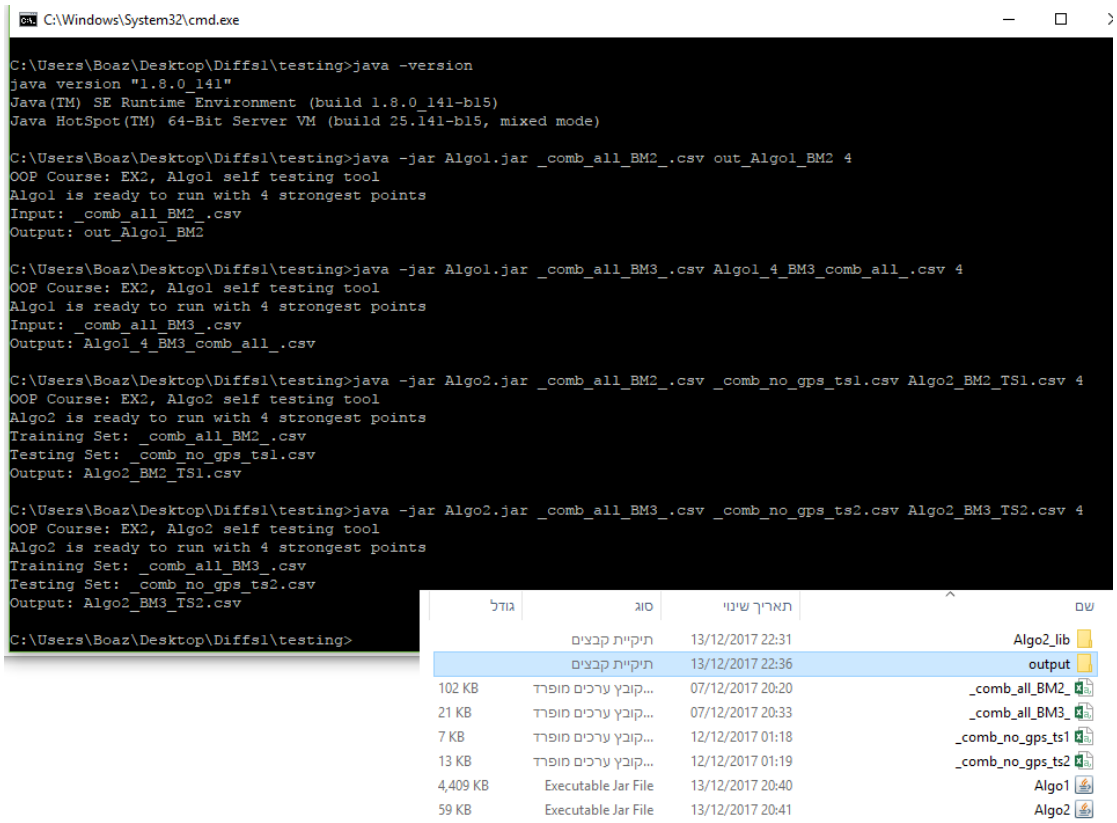
Algo1: java -jar Algo1.jar <input_file> <output_file> <number_of_sampls>
java -jar Algo1.jar _comb_all_BM3_.csv Algo1_4_BM3_comb_all_.csv 4

Algo2: java -jar Algo1.jar <training_set> <testing_set> <output_file> <number_of_sampls>
java -jar Algo2.jar _comb_all_BM3_.csv _comb_no_gps_ts2.csv Algo2_BM3_TS2.csv 4

3.3 עבור אלגוריתם 1, הכינו טבלת השוואה ב excel בין המיקום שאתם קיבלתם למיקום שהאלג' שסופק לכם חישב – התייחסו לשוני האופייני (הממוצע) בין המימוש שלכם לזה שסופק לכם.

3.4 עבור אלגוריתם 2, הכינו טבלת השוואה ב excel בין המיקום שאתם קיבלתם למיקום שהאלג' שסופק לכם חישב – התייחסו לשוני האופייני (הממוצע) בין המימוש שלכם לזה שסופק לכם.

3.5



רכזו את כל החומר בדוח מסודר שתעלו ל github שלכם – את הדוח יש לשים בתיקייה בשם docs.

הערות:

- הפתרון שניתן לכם נבדק בחלונות וב linux בגרסת java1.8 – תיתכן חוסר התאמה לגרסאות ישנות של java. בכל מקרה ניתן לעשות את המטלה גם בלי להריץ את הפתרון שכן סיפקנו לכם את קבצי הפלט.
- פתרון חלקי למטלות 0,1 וכן הדרכה למטלה 2 נמצאים בקישור הבא:
https://github.com/benmoshe/OOP_Exe - מותר לעשות שימוש בקוד מכאן או מכל מקור אחר ברשת אבל חייבים לציין זאת בפירוש!