הוראות לתרגיל "בית חכם"

קורס: IOT

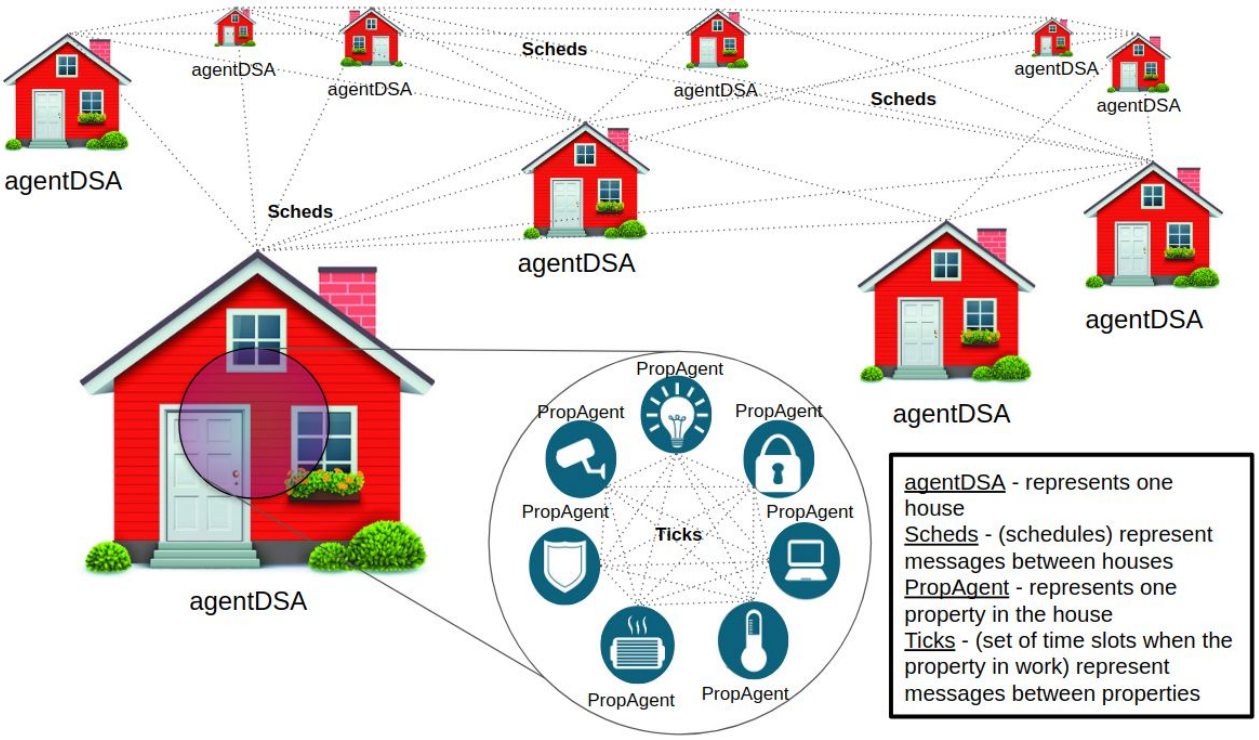
1.תיאור מטלה

ברוכים הבאים לשכונה חדשה של בתים חכמים בבאר שבע. בשכונה כל בית מתקשר עם כל בית אחר, בכל בית יש מכשירי חשמל שגם הם מתקשרים זה עם זה. הוזמנתם כמהנדסי תעשיה וניהול עם ידע מעמיק באינטרנט של הדברים, בכדי לעזור לצוות אחזקה של השכונה לפתח אלגוריתם חדש שבעזרתו תתאזן צריכת החשמל השכונתית וכיוצא מכך יופחתו חשבונות החשמל של התושבים.
לאחר ניתוח המצב הקיים, התגלה לכם שהאלגוריתם המשמש את הסוכנים המפעילים את המכשירים בבתים הוא אלגוריתם חמדני פשוט (greedy) ובתקשורת בין המכשירים יש חילוף מידע מינימלי (איטרציה אחת) והפתרון נבחר כמעט רנדומלית. קיבלתם המלצה לממש אלגוריתם DSA בפתרון הבעיה. בפניכם שתי אופציות (עליכם לבדוק את שתיהן):

1. לממש את האלגוריתם כשכל **בית** הוא סוכן. הבית מחליט באופן מרוכז על ההשמות של המכשירים שלו. בכל איטרציה: הבית מקבל השמות של הבתים השכנים. הבית דוגם 10 פתרונות רנדומליים אפשריים חדשים (כל פתרון הוא אוסף השמות עבור כל מכשיר בבית). הבית משווה ביניהם, ולוקח את הפתרון הטוב ביותר (השוואה מתבצעת באמצעות פונקציה ייחודית הבודקת את צריכת החשמל בשכונה). במידה והפתרון החדש טוב יותר מהפתרון הקיים או שווה לו, הבית עובר אל הפתרון החדש בהסתברות 0.7. במידה והפתרון החדש רע יותר מן הפתרון הקיים, הבית נשאר עם הפתרון הקיים. בסוף האיטרציה, הבית שולח לכל השכנים את הפתרון הנבחר.
יש לבצע 50 איטרציות ולהוציא גרף המצביע על איכות הפתרון של האלגוריתם.

2. לממש את האלגוריתם כשכל **בית** הוא סוכן ובתוך בית כל **מכשיר** הוא סוכן גם כן (כל מכשיר בבית הוא PropAgent). לכל מכשיר בבית יש רשימת השמות אפשריות (באיזה זמנים המכשיר יכול לעבוד בהינתן אילוצים קיימים).
שימו לב: בהמשך מדובר על שני סוגי איטרציות. ישנן איטרציות פנימיות שרצות בתוך הבית וישנן איטרציות הגלובליות בין בתים כמו באופציה 1.
יש לממש את DSA **ברמת המכשיר**, יש לבצע 100 איטרציות פנימיות בתוך כל איטרציה גלובלית. בכל איטרציה גלובלית: הבית מקבל השמות של בתים שכנים. לאחר מכן, עובר לאיטרציות פנימיות. בכל איטרציה פנימית: כל מכשיר מקבל השמות של המכשירים האחרים בבית. על המכשיר לבחור את ההשמה המועדפת מרשימת השמות אפשריות שיש לו.
במידה וההשמה שבחר משפרת את צריכת החשמל בשכונה (ישנה פונקציה ייחודית הבודקת את צריכת החשמל בשכונה) יש ולעבור אליה בהסתברות של 0.7. במידה והיא לא משפרת את צריכת החשמל בשכונה יש להישאר עם השמה קודמת.
בסוף ה-100 האיטרציות הפנימיות הבית מעביר למכשירים את ההשמות שנבחרו, שולח לבתים שכנים וממשיך לאיטרציה הגלובלית הבאה.
יש לבצע 50 איטרציות גלובליות ולהוציא גרף המצביע על איכות הפתרון של האלגוריתם.

לאחר מימוש שתי האופציות הנ״ל אתם מתבקשים להריץ על **אותה קבוצת בעיות**, להשוות ביניהן ולהציע את הפתרון המועדף לדעתכם. נא לספק נימוק משכנע לבחירתכם.
מהנתונים שנאספו התברר כי מבנה התקשורת בשכונה הוא כדלקמן (האיור בעמוד הבא):



עתה נכיר סביבת סימולציה בה תוכלו לבדוק את העבודה שלכם שכוללת קבצי קוד המנהלים את הבתים בשכונה. התושבים מאמינים בכם ומאחלים המון בהצלחה.

2.סביבת סימולציה

סביבת סימולציה בה תעבדו יכולה להיות מותקנת על שני סוגי מערכות הפעלה: לינוקס (אובונטו) או ווינדוס.

אפשרויות התקנה:

א. לינוקס - אתם יכולים להתקין מערכת הפעלה זו לצד מערכת הפעלה שיש לכם במחשב ולאחר מכן לגשת ל**מדריך למשתמש** של הסימולטור כדי לסיים את התקנת התוכנה. שני תהליכי התקנה אלו פשוטים ולא דורשים זמן רב. כדאי להיצמד להוראות בעמוד הרשמי של אובונטו בזמן ההתקנה.

ב. ווינדוס - לגשת ישירות להוראות התקנה ב**מדריך למשתמש** של הסימולטור ולסיים את התקנת התוכנה.

ג. להגיע למעבדה 48 בה מותקנת התוכנה.

ב**מדריך למשתמש** הנמצא באתר הקורס תוכלו למצוא את כל המידע הנדרש להיכרות ראשונה עם סביבת סימולציה. בנוסף לכך ייפתח פורום לשאלות בנוגע לפרויקט ולסימולטור. נא להפנות את כל השאלות הרלוונטיות לפורום זה.

בנוסף לכך תקבלו קובץ ג'אווה, שעליו תממשו את האלגוריתם שלכם ובהמשך תוכלו לטעון אותו לתוך התוכנה לצורך בדיקת הפתרון שלכם.

3.קבצי HouseProp ו- House

כל הקוד שתכתבו חייב להיות בשני קבצים (אחד לכל פתרון) והם קבצי ג'אווה ספציפיים שנקראים **House.java ,HouseProp.java**.
במידה ותרצו להוסיף עוד אובייקטים או פונקציות משלכם תוכלו לעשות זאת באמצעות (אם מדובר על אובייקטים) מחלקות פנימיות או (אם מדובר על פונקציות) פשוט לכתוב אותם בפנים. דוגמא:

```
public class House extends SmartHomeAgentBehaviour {
        ...
        myFunc1(){
        ...
        }
        myFunc2(){
        ...
        }
        ...
        class PropAgent {
                ...
        }
        class MyClass1{
                ...
        }
        class MyClass2{
                ...
        }
}
```

ניתן ליצור כמה מחלקות פנימיות שתרצו אך חשוב לא לתת להם הרשאות public, כי המחלקות הנוספות האלה נועדות רק לצורך שימוש פנימי.

כמו שהוזכר מקודם בשכונה מתבצעים שני תהליכים במקביל:

1. בתים שולחים אחד לשני את הלו"זים (schedules) שלהם ליממה הקרובה.
2. מכשירים מתעדכנים אחד מהשני על השעות (Ticks) שהם פועלים ביממה הקרובה.

עתה נראה כיצד התהליכים האלה ממומשים בקוד:
החלפת מסרים (לו"זים) בין בתים מתבצעת בשיטה doIteration() באמצעות פונקציה receiveNeighboursIterDataAndHandleIt().
לאחר מכן בתוך כל בית בנפרד יש שיטה improveSchedule() בה כבר המכשירים מחליטים על השמה שלהם ו**בה תממשו את הפתרון שלכם**.
ולבסוף ההשמות שנבחרו על ידי המכשירים מתרכזות בלו"ז חדש של בית (updateTotals()) והלו"ז הזה נשלח לכל הבתים האחרים (מתבצע מאחורי הקלעים של סימולטור).

נא לעבור בעיון על קטע קוד וההארות המלוות אותו כדי להבין סדר פעולות דרוש להצלחת אלגוריתם.

4.הערות כליות

- נגיד וטענתם את האלגוריתם שלכם לתוכנה, אבל אחר כך שיניתם בקוד משהו ורוצים לטעון אותו מחדש, אז מה שתצטרכו לעשות זה למחוק את קבצי הclass שנוצרו עבורו בתוך התקיה של התוכנה. כנסו לפה:

ותמחקו את הקבצים שנוצרו עבור האלגוריתם שלכם. תסגרו ותכנסו לתוכנה מחדש ורק אז תוכלו לטעון שוב את האלגוריתם המעודכן שלכם.

- שפת תכנות: Java
- מומלץ מאוד לעזור אחד לשני בהבנה של התרגיל וקבצי הקוד הרלוונטים. יחד עם זאת לא תתקבלנה העתקות ונתייחס אל אירועים כאלה בחומרה.
- לוגר (logger) - מדפיס את המידע בטרמינל שנפתח בהפעלת התוכנה. תוכלו להשתמש בו כדי לדבג את הקוד שלכם. שימו לב שקיימות עוד הדפסות רבות לאותו טרמינל מהתהליכים הפנימיים של התוכנה ולפעמים צריך לגלגל הרבה למעלה כדי למצוא את ההדפסה שיצרתם.
- בקובץ ג'אווה יש הרבה דברים פחות רלוונטים לביצוע תרגיל, שם גם תמצאו פחות הסברים בהערות. כמו כן, במקומות שיש בהם הרבה הערות כדאי להתעמק יותר ולהבין עד הסוף.

5.סדר פעולות מומלץ

אנחנו ממליצים את סדר הפעולות הבא. כך תוכלו להתקדם בהדרגתיות ולהימנע מטעויות מיותרות.

1. התקנת תוכנה - תבדקו שהכל רץ בלי שגיאות, העמוד הרלוונטי נפתח בדפדפן והאלגוריתמים המובנים רצים בלי התראות.
2. תגיעו להרצאת הסבר על התרגיל. (נפרסם שעה ומיקום באתר הקורס)
3. תורידו את הקוד של סימולטור בקישור שנמצא במדריך למשתמש ותפתחו אותו ב**InteliJ Idea**. התוכנה הזאת חינמית
4. תנסו להכניס אלגוריתם חדש באמצעות טעינת קובץ House ו- HouseProp לתוכנה ותריצו אותו. מהתוצאות שתקבלו בגרפים ניתן יהיה לראות שהאלגוריתם לא יעיל. זה בדיוק הדבר שאתם הולכים לשפר.
5. תעיינו בקוד בקבצים הבאים: SA, HouseProp, House, בהם תוכלו למצוא הסבר מפורט על המרכיבים העיקריים שבהם תשתמשו. נא לעבור עליהם בקפדנות ובסבלנות, ככל שתשקיעו מאמץ רב יותר בשלב זה כך תעבדו פחות קשה בהמשך.
6. תחשבו על דרכי פתרון אפשריים. חשוב כאן לציין כי קיימות דרכים רבות לפתור את הבעיה הנתונה, לכן בבדיקה ניתן משקל לשיקולים בבחירת הפתרון.
7. השתמשו בתוכנה כדי לבדוק את התוצאות שהאלגוריתם שלכם משיג. במידה וקיימת בעיה כלשהי באלגוריתם חשבו על דרכי פתרון אפשריים, היעזרו בחברים שלכם, במידע באינטרנט או בפורום במודל.
8. השוו בין שתי האופציות המתוארות בחלק הראשון של התרגיל באמצעות גרפים שהתוכנה פולטת. מה לדעתם האופציה המועדפת? תנסחו את תשובתכם בדוח התרגיל עם נימוקים מפורטים.
9. הגישו אלגוריתמים סופיים והדוח באופן הבא: קובץ zip המכיל בתוכו קבצי ג'אווה בשם HouseProp ו- House וקובץ PDF עם הדוח והסברים נוספים שיעזרו לנו להבין את הקוד ואת הסבר קצר על הפתרון שהצעתם (לא יותר משלושה עמודים). שם של קובץ zip הוא ת.ז. של המגיש. בדוח נא לציין ת.ז. של המגיש גם כן.