**אלגוריתם 2 - הבנת הרשת:**

נצא מנקודת הנחה שרובוט X כבר יודע מי הם שכניו.

ביכולתו של רובוט X לשדר לשכניו מעין "טבלה" (רשימה מקושרת) ובה מידע על שכניו ובצורה כזו להסתנכרן עם השכנים שלו ע"מ שיכירו אחד את השני.

בצורה הזאת כל רובוט יוכל להכיר את כל הרובוטים ב"זירה", ומעבר לכך כל רובוט יוכל לחשב את מסלולי התקשורת המועדפים לכל אחד מהרובוטים השונים ב"זירה".

הרובוט X בודק שהוא יכול לשדר את ההודעה, במידה והוא יכול ואין עוד הודעה שמשודרת באותו פרק זמן הוא מפיץ לשכניו את ההודעה.

השכנים עוברים על הרשימה (עד isEmpty ) ומעתיקים את זה לרשימת המידע שלהם.

כך רובוט יכול להכיר את הסביבה שלו (פונקציית Arena ).

מרחק בין הרובוטים הוא גם המסלול הכי קצר. נשתמש בנוסחה המוכרת לחישוב מרחק (נשתמש **באלגוריתם דייקסטרה** ע"מ שנוכל להתחשב גם במכשולים).

{

נעבור על כל הרשימה ונשמור בתוך משתנה C את האיבר הראשון במחסנית.

נרוץ בfor על השכנים, כל שכן עובר על הרשימה (עד isEmpty ) .נבדוק בif אם השכן הוא מינוס 1 . אם כן: C+1.

ונכניס למחסנית את כל המידע אצל אותו השכן.

}

**אלגוריתם 3 – שיערוך המיקומים:**

נניח שמפוזרים על הזירה רובוטים "סטטיים" (שלא זזים – "עצים"). הרובוטים ה"סטטיים" יודעים את מיקומם ע"ג הזירה.

באלגוריתם זה ננסה לפתח שיטה בעזרתה נשערך את מיקום שאר הרובוטים לפי עוצמת הסיגנל הנקלט ושימוש ברובוטים הסטטיים כעוגנים לחישוב:

נוכל לבצע **סטית** תקן (deviation) של הרובוטים ע"ג הזירה ע"י סכימה של הממוצע של המרחקים של הרובוטים (בריבוע).

בשיטה זו יכול הרובוט להבין שאם הוא קלט הודעה מאותו הרובוט הסטטי זה אומר שהוא בתוך הטווח שלו, הוא יוכל לבצע סטית תקן עם האחרים וכך לשער את המיקום שלו.

{

נשלח כל פעם לפונקציה שתבדוק מרחק את הרובוט ואת הנקודה שלו (הX וה- Y של הנקודה ) ונשמור את זה בתוך משתנה (Distance) שכמובן צריך להיות גדול או שווה ל0.

(במידה והוא שווה ל0 פשוט נוסיף אותו למשתנה שיסכום את המרחקים הקצרים).

ולבסוף נחלק במספר הרובוטים ע"מ לקבל ממוצע.

}