**מבוא לבינה מלאכותית**

**סמסטר חורף תשפ"ב**

**מטלה 1**תאריך הגשה: 14.11.21 23:55

# הנחיות

* שאלות בנושא מטלה זו יש לשאול דרך המודל, בפורום "מטלה 1".
* הוראות להגשת המטלה מופיעים בסוף מסמך זה.
* הקבצים הנדרשים להרצת הקוד הינם:
  + a\_star\_search.py
  + breadth\_first\_serach.py
  + main.py
  + route\_grid.py
  + grid\_1
  + grid\_2
* העבודה להגשה ביחידים בלבד אלא אם כן המגישים קיבלו אישור להגשה שאינה ביחידים.
* לפני שניגשים לממש את המטלה מומלץ לעיין רבות בהסברים וכן בקוד הקיים.
* פתרון המטלה שתגישו ייבדק מול שאר ההגשות על ידי תוכנת העתקות.
* **מי שימצא כי העתיק יכשל בקורס וכן יועבר לוועדת משמעת אוניברסיטאית**.
* הפרויקט נכתב וייבדק בשפת התכנות python, גרסה 3.6 עד 3.8.
* יש להתקין את החבילות pandas 1.1.4 ו- numpy 1.19.3.
* מסמך זה בנוי באופן הבא: תיאור המטלה, בעיית החיפוש, מרחב הבעיה, שאלות המטלה, פונקציות שיש להשלים והסבר על פונקציות קיימות.

# תיאור המטלה

יצרנית רכבים אוטונומיים מעוניינת לנסות מספר אלגוריתמי חיפוש למציאת מסלול להגעה ליעד. כל אחד מהמתכנתים בחברה בחר להציג אלגוריתם שונה, כאשר קובי ביקש להציג את האלגוריתם**Breadth-first search (BFS)** ורוני הציג את האלגוריתם **\*A**. במהלך סופת הברקים האחרונה נמחקו הגרסאות האחרונות, ולכן עליכם לעזור לקובי ורוני להשלים את הפונקציות שלא נשמרו.

# בעיית חיפוש

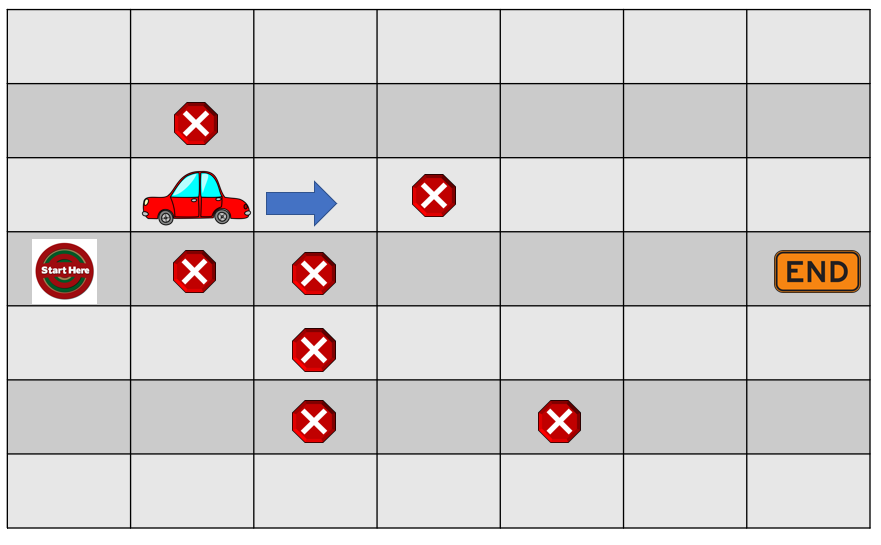
בהינתן מיקום הרכב האוטונומי ומיקום יעד רצוי על גבי מפה, עלינו למצוא מסלול עבור מכונית אוטונומית כך שהמכונית תוכל להסיע אותנו מהמיקום ההתחלתי אל מיקום היעד.

# מרחב הבעיה

גריד שמיוצג ע"י מטריצה בגודל , כאשר נתונים התאים של המצב ההתחלתי והמצב הסופי. הגריד הינו זהה לשני האלגוריתמים ומיוצג על ידי מערך דו-מימדי.

מצב: מיקום של המכונית האוטונומית.

אופרטור: התקדמות בכיוונים צפון, דרום מזרח או מערב. כמו כן, לא ניתן להתקדם למשבצת "חסומה".



# שאלות המטלה

1. עזור לקובי להשלים את הפונקציות החסרות (מצויינות בהמשך) בקובץ **breadth\_first\_search.py**, כך שהחיפוש שימומש יהיה אלגוריתם חיפוש לרוחב (BFS). מצב בגרף החיפוש (state) עבור אלגוריתם BFS מיוצג על ידי אובייקט python מסוג tuple המכיל את הערכים הבאים: מיקום x, מיקום y וה-state ממנו הגענו (המצב ההתחלתי מכיל False כערך ה-state). הפונקציה של החיפוש (בשם bfs) הינה ממומשת ומחזירה את ה-state הסופי עם הדרך המלאה ליעד.
2. עזור לרוני להשלים את הפונקציות החסרות בקובץ **a\_star\_search.py**, כך שהחיפוש שימומש יהיה אלגוריתם A\*. מצב בגרף החיפוש (state) עבור אלגוריתם BFS מיוצג על ידי אובייקט python מסוג tuple המכיל את הערכים הבאים: ערך f (ערך משוקלל של g+h), ערך h, ערך g, מיקום x, מיקום y, ה-state ממנו הגענו (המצב ההתחלתי מכיל False כערך ה-state). הפונקציה של החיפוש (בשם: astar\_search) הינה ממומשת ומחזירה את ה-state הסופי עם הדרך המלאה ליעד.

# הפונקציות שיש להשלים למימוש כל אחד מאלגוריתמי החיפוש

· init\_open():

הפונקציה אינה מקבלת ערכים, ומחזירה מבני נתונים מתאים לאלגוריתם החיפוש.

· insert\_to\_open(open\_list, s):

הפונקציה מקבלת כקלט את מבני הנתונים open ומצב s ומכניסה את s ל-open.

· get\_best(open\_list):

הפונקציה מקבלת כקלט את מבני הנתונים ומחזירה את הקודקוד הטוב ביותר, על פי אלגוריתם החיפוש.

· get\_neighbors(grid, s\_location):

הפונקציה מקבלת את המרחב (grid) ואת המיקום הנוכחי s\_location, ומחזירה את כל מיקומי השכנים המתאימים להתקדמות (לאחר סינון המצבים שאי אפשר להתקדם אליהם – מחוץ ל-grid או חסומים).  
מיקומים חסומים מיוצגים ב-grid על ידי התו '@'.

· is\_goal(s\_location, goal\_location):

הפונקציה מקבלת את המיקום הנוכחי ואת המיקום המטרה ומחזירה האם הגענו למטרה.

· calculate\_heuristic(s\_location, goal\_location):

(קיימת רק ב-\*A) הפונקציה מעריכה את המרחק מהמיקום הנוכחי למטרה ומחזירה את הערכה זו.

# הסבר על הפונקציות הקיימות

get\_route(s):

פונקציה המקבלת את ה-state הסופי מהפונקציות bfs \ astar\_search והופכת אותה לרשימה נוחה להדפסה ו-debugging.

print\_route(s):

פונקציה המקבלת דרך סופית (ע"י פונקציה get\_route) ומדפיסה את הדרך המלאה עד עליה.

astar\_search/bfs(grid, start\_location, goal\_location):

פונקציה המממשת את אלגוריתם החיפוש (המתאים), מקבלת grid כמטריצה בגודל (המימדים אינם ידועים מראש), מיקום התחלתי ומיקום סופי.

# הגשת המטלה

* יש להגיש דרך מערכת המודל את שני הקבצים הבאים:
  + a\_star\_search.py
  + breadth\_first\_serach.py

הגשת המטלות תתבצע ישירות מול מערכת המודל בצורה אלקטרונית.

שימו לב: ישנה אפשרות להגיש את המטלה התכנותית מספר פעמים ובכל הגשה לקבל חיווי, כלומר תקבלו באופן מיידי את הציון לביצוע התרגיל.

ניתן לראות את הפידבק להרצה (קומפילציה, מספר טסטים שעברו, שגיאות זמן ריצה וכו'...).

לאחר סיום ההרצה יתקבלו התוצאות. ישנם 4 טסטים הבודקים את הפתרון המוגש למטלה. אם התקבלה שגיאת קומפליציה יש עליכם להעלות קובץ חדש בכדי לקבל ציון לאחר התיקונים. בקבצים המקוריים שקיבלתם עם המטלה ישנם שגיאות ריצה עבור פונקציות לא ממומשות.

בהצלחה 😊