**פרויקט סיום בקורס מבוא לבינה מלאכותית 67842**

# משחק הוויקיפדיה

מאיה כהן 203821707 | אלון רכס 305765026 | אלה נאמן 204106918

משחק הוויקיפדיה, המוכר גם בשם Wiki-Race הוא משחק פשוט ופופולרי במסגרתו נדרשים המשתתפים לשוטט בין הערכים השונים באינציקלופדיה האינטרנטית. מטרתם היא ליצור מסלול מערך מקור לערך יעד המוסכמים על כלל השחקנים, ע"י לחיצה על קישורים בתוך הערכים ולא ע"י שימוש במנועי חיפוש או קישורים פנימיים בתוך וויקיפדיה. הגדרתו של מסלול טוב תלויה בגרסת המשחק ובאסכולה אליה משתייכים השחקנים המשתתפים: לרוב מדובר במסלול שנמצא תוך זמן קצר או כזה שנמצא ראשון מבין המתחרים, לעיתים מדובר במסלול שכולל מספר קטן ככל הניתן של דילוגים, ויש שמחפשים גם מסלול שעושה שימוש מינימלי בתעבורת רשת נצרכת בזמן החיפוש.

בחרנו למדל את וויקיפדיה כגרף מכוון שקודקדיו הם הערכים וצלעותיו הן לינקים חד-כיווניים. בעיית מציאת מסלול קצר ביותר על גרף מנקודה לנקודה היא בעיה קלאסית פתורה (דייקסטרה), ומחשבים עולים ביכולתם עשרות מונים על בני אדם בפתרונה. אולם מאפייניה היחודיים של וויקיפדיה הופכים את הבעיה לכזאת שמתאימה לעולם הבינה מלאכותית: לעומת חיפוש באפלה שמבצע מחשב, בני אדם מגיעים עם ידע מוקדם נרחב, יכולות עיבוד שפה ותמונות באופן אינטואיטיבי ומיידי. הם מצליחים לנווט בגרף, להתחקות אחרי מסלולים לא טריוויאלים, ולבחור בתבונה מאיזה ערך כדאי להתקדם ואילו ערכים כדאי לזנוח.

כמה מתכונותיה של וויקיפדיה המעניקות יתרונות משמעותיים לאדם בביצוע המשימה על פני מחשבים:

- **דינאמיות:** וויקיפדיה היא אתר דינאמי המתעדכן ומשתנה כל העת. קיימים אלגוריתמים קלאסיים המתבססים על ניתוח ומיפוי המרחב מבעוד מועד, ע"י ביצוע עיבוד מוקדם. אלגוריתמים אלו לא יכולים למצוא פתרון סופי וסגור לבעיה: אפילו שינוי בודד עשוי ליצור

קיצור דרך שימושי בין תחומים רחוקים ובכך להשפיע על כמות גדולה של מסלולים, כך שמידי יום-יומיים הוא מפסיק לספק תשובות אופטימליות. כלומר, העיבוד המוקדם צריך להוסיף לעבוד ולהשתנות ללא הרף. אם מדובר בעיבוד מקדים יקר האלגוריתם כולו בזבזני.

- **גודל וקישוריות:** וויקיפדיה היא אסופת מאמרים בהיקף חסר תקדים. וויקיפדיה האנגלית כוללת יותר מ-5.5 מיליון ערכים. ממוצע הלינקים בעמוד הוא 77, כאשר ללפחות אחוז אחד מן הערכים יש יותר מ-800 לינקים יוצאים שונים, ולאחוז אחד יש יותר מ-2,300 לינקים נכנסים. יתר על כן, בחירת ערכי התחלה וסיום בשיטות שונות הובילו אותנו כמעט תמיד למצוא מסלולים מפתיעים באורך 2-6 דילוגים, ומסלולים בודדים שדרשו 7 דילוגים לכל היותר. מחד, לכאורה קיימים מסלולים קצרצרים ועל כן מציאתם נראית קלה. מאידך, אם נניח שרוב הערכים קשורים ביניהם בשבעה דילוגים לכל היותר, נסיק שהמאגר ניחן בקישוריות בעלת אופי מבוזר מאד, כלומר מידת ההסתעפות של מסלולים היא אדירה, ומגוון הערכים שניתן להגיע אליהם תוך 5-6 דילוגים הוא עשיר מאד. אפילו בהנתן יכולות חישוב זריזות וזיכרון בלתי נדלה, שיטוט קלאסי ללא הכוונה בגרף כזה היא פעולה יקרה וקשה, לעומת גלישה עירנית ומכוונת מטרה.
- **סידור פנימי אינטואיטיבי:** אם נתעלם מאינדקסים פורמליים וממנוע החיפוש של וויקיפדיה, נישאר עם מרחב ערכים בעל סידור פנימי מקרי למדי. אם ננסה לאפיין מרחק בין ערכים ע"י מספר הלינקים המקשרים ביניהם, לא נקבל מטריקה. זאת משום שהפניות בין ערכים בדר"כ אינן סימטריות.
    - בקטגוריית אישים למשל, לכל דמות לרוב מצויינת ארץ המוצא, אך לא כל הדמויות הבריטיות מאוזכרות בערך הממלכה.

מלבד זאת, מאד נפוצים "קיצורי דרך" או מטאפורות שמובאות כדוגמא או כהמחשה, אך הקשר בינן לבין הערך הוא מקרי למדי. בפרט, וויקיפדיה נוטה להכיל מגוון של עובדות איזוטריות בערכיה השונים עקב האופי השיתופי והפירוט הנרחב אודות ערכיה.

- למשל, "סלט פירות" בוויקיפדיה העברית מפנה ל"מיונז","חרדל" ו-"חסילונים" אך לא ל"בננה" או לאף פרי מתבקש אחר.

מובן שקיים איזשהו היגיון כללי ואינטואיטיבי והמרחק המתואר אינו רנדומי כלל, אבל קשה לאפיין או לחזות את החוקיות שבבחירת דוגמא ספציפית כזו או אחרת להסבר מסויים, והייצוג שמקבלות עובדות שוליות אינו תואם את מידת הקשר שלהם לנושא הערך. על כן, הניסיון "ללכת בכיוון הערך" הוא די חמקמק וקשה לאפיון.

### חיפוש לא אדמיסבילי

על פניו, אין תכונה קלה לזיהוי המרחק בין קודקודי הגרף. כשבחנו את הבעיה חשבנו מיד לפתור אותה ע"י מציאת יוריסטיקות לחיפוש באמצעות אלגוריתם A*. אך מכיוון שהערכים נכתבים בספורדיות, ע"י כותבים מנוסים ולא מנוסים, תחת פורמט קבוע, אבל לא באופן מפוקח לחלוטין, מהר מאד הבנו שקשה מאד לחזות כיצד הם יראו, ואין אחידות בינהם. על כן, הבנו שקשה עד בלתי אפשרי למצוא יוריסטיקה שתתן תמיד שערוך אופטימי לערך. למרות זאת, מצאנו כמה יוריסטיקות מרשימות שמשפרות את החיפוש בפועל, למרות שהן אינן אדמיסביליות.

- למשל, יוריסטיקת כמות הלינקים, שמשיגה תוצאות יפות יחסית (בפרק התוצאות), סופרת את כמות הלינקים היוצאים מהאב, את כמות הלינקים היוצאת של בניו, ומחזירה את המנה שלהם. מתוך כוונה שהאלגוריתם ייטה לחפש בערכים עשירים יותר בקישורים. יוריסטיקה זו איננה אדמיסבילית. לדוגמא, אם נביט במסלול קצר ביותר (אין מסלולים עם צלע אחת) בין הערכים "Fruit salad" ו-"Chicken"

Fruit salad → Waldorf salad → Chicken

היוריסטיקה נותנת ציון לא טוב (גבוה) לערך Waldorf salad, כי הוא ערך עם פחות מ-30 לינקים יוצאים, ובוודאי מעדיפה על פניו (ציון נמוך) את הערך Philippines, ובו מאות קישורים, וגם הוא לינק שמופיע ב-"Fruit salad".

## פרטי המימוש

בהתחלה ניסינו לעבוד רק מול ה-API של וויקיפדיה בעזרת ספריית wikipedia של Python. אחרי ניסיונות הרצה ראשוניים ראינו שקצב התקשורת איטי וזמן הריצה מאד ארוך גם במקרים פשוטים. הצענו כמה פתרונות לבעיה:

- הרחבנו את ספריית wikipedia שתאפשר לתשאל מידע על מספר דפים בבת-אחת (ההרחבות נמצאות תחת תיקיית improved_wikipedia בקוד).
- הרצנו שאילתות ב-threads נפרדים
- הורדנו מסד-נתונים המכיל את כל הקישורים בין ערכי הוויקיפדיה[1].

לכל אחת מהשיטות היו יתרונות וחסרונות: תשאול ה-API עם מספר רב של דפים העלה משמעותית את קצב הריצה, אך ככל שרצינו לקבל יותר מידע עבור כל דף כמות הדפים שניתן היה לשאול לגביהם ירדה. שימוש ב-threads הביא את וויקיפדיה לחסום אותנו אחרי מספר דקות או שעות של ריצה (כתלות בכמות), כך שלבסוף נאלצנו לוותר ולהשאר עם thread בודד. שימוש במסד הנתונים הוא מהיר מאוד, אך הוא לא נותן עוד מידע מלבד קישורים.

בחלק מהיוריסטיקות מספיק היה לעבוד מול מסד הנתונים (אלגוריתמים offline) וחלקן עובדות מול הAPI עם improved_wikipedia (אלגוריתמים online).

## חיפוש דו-כיווני

המוטיבציה למימוש חיפוש דו-כיווני הייתה חיסכון משמעותי בזמני ריצה. רצינו להריץ שני אלגוריתמי A* במקביל, אחד שיתחיל מערך ההתחלה וישתמש בלינקים יוצאים, והשני שיתחיל מערך היעד וישתמש בלינקים נכנסים. מטרתם היא להיפגש או לנווט אל עבר קודקוד משותף, כך שיווצר מסלול אחד חוקי משני חלקי המסלולים שיווצרו. כך, במקום לחפש בעץ עם עומק d, מחפשים במקביל בשני עצים עם עומק d/2, ומבקשים לצמצם משמעותית את זמני הריצה ולשפר את הביצועים.

---

[1] מסד הנתונים (עדכני ליולי 2018):
https://drive.google.com/open?id=1kf_FEVSy6z_ACcL7kqop9a6LCXAP8jKB

בפועל, החיפוש הדו-כיווני משיג זמני ריצה שעומדים במשימה, בעוד החיפוש החד-כיווני לא מגיע לתוצאות מספקות תוך מספר דקות.

השירותים שוויקיפדיה מספקת תחת ה-API שלה אפשרו לנו לממש את החיפוש הדו-כיווני בגרף האינציקלופדיה האינטרנטית. וויקיפדיה מעדכנת רשימה של לינקים נכנסים לערך, שנקראת " what links here". גם במאגר הנתונים ה-offline סופקה לנו רשימה של לינקים נכנסים. חשוב להבחין כי הלינקים הנכנסים והלינקים היוצאים הם אינם אותה קבוצה, ולכן לו לא הייתה נתונה לנו רשימה כזאת לא ניתן היה לבצע את החיפוש במקביל.

הרחבת אלגוריתם A* לביצוע משימות חיפוש דו-כיווניות אינה משימה טריוויאלית. כדי לנווט בהצלחה אל עבר "קודקוד המפגש" היוריסטיקה צריכה לדעת להעריך מרחק מקודקוד נוכחי אל קודקוד המפגש, אך זאת מבלי לדעת במפורש מיהו הקודקוד הזה. כדי בכל זאת לבצע חיפוש דו-כיווני יעיל הצענו פרשנות לאיך לכתוב ולממש אלגוריתמים אלה תוך כדי שימור עקרונות A* והכללתם.

- **A* דו-כיווני דינאמי:** שני אלגוריתמים שפועלים במקביל, מהמקור ליעד ומהיעד למקור, המחזיקים כל אחד פרינג' משלו. כל קודקוד עוקב באלגוריתם הישר (מקור←יעד) מוערך ע"י יוריסטיקה שמחשבת את מידת הקרבה שלו אל הפרינג' של האלגוריתם ההפוך (יעד←מקור), כלומר אל מקבץ קדוקדים ולא אל קודקוד יעד אחד. לאחר מכן, עוברים על הפרינג' ההפוך ומבצעים הערכה מחודשת של כל חוליה בו ביחס למקבץ הישר המעודכן. בעבודה זו החלטנו שלא לממש או לעסוק בגרסה הדינאמית, היות והערכנו שדרישות זמן הריצה שלה יהיו מוגזמות בגלל העדכון התכוף של הפרינג'ים, ומיונם מחדש.
- **A* דו-כיווני סטאטי מונוטוני:** אלגוריתם זה מניח שיש איזשהי תכונת "כיוון" או "סדר" הגיונית ומונוטונית על המרחב, כלומר שהתקרבות אל ערך היעד מבטיחה גם התקרבות אל הקודקוד המשותף. במצב כזה אין צורך להסתבך עם חיפוש והגדרת יעדי ביניים מורכבים ואפשר פשוט לנסות לחתור אל עבר היעד הידוע. בגרסה זו מריצים את האלגוריתם הישר וההפוך באופן בלתי תלוי, לסירוגין, ותמיד מודדים את המרחק מאותו קודקוד. הפונקציה

שבודקת אם החיפוש הסתיים מבצעת חיתוך בין הפרינג'ים של שני האלגוריתמים ואם היא מוצאת ערך מסויים בחיתוך של שניהם סימן שהוא קודקוד משותף. השתמשנו בו עבור יוריסטיקות מעולם עיבוד השפה הטבעית.

- **A* דו-כיווני סטאטי אוניברסלי:** אלגוריתם זה מנסה לאתר קודקוד מפגש שאינו תלוי בקדוקד היעד או בקודקוד המקור, או בהתקדמות החיפוש בשני הצדדים. הדבר דומה לקביעת נקודת מפגש מראש וניסיון להגיע אליה משני צדדי החיפוש. במקרה כזה המשחק משתנה והשאיפה שלנו היא לתת דירוג אוניברסלי לכל ערך. אלגוריתם זה נתפר במיוחד עבור המודל של וויקיפדיה, בלא הנחת סדר על ערכיה. להערכתנו, ישנן קליקות של ערכים פופולריים דומים שהמעבר ביניהן מתבצע בקלות. היורסיטיקות שמיועדות לגרסה זו צריכות לקדם ערכים שמובילים לגידול משמעותי במידת השייכות-לקליקה-עשירה ולזנוח ערכים שאינם חלק מקליקה מרשימה, או שכבר יש אחיזה בקליקה שלהם. צפינו שאלגוריתם זה יתקשה למצוא מסלולים באורך אופטימלי, אבל האמנו שהזהירות מחזרה על ערכים באותה קליקה תוביל ליתרון משמעותי בגזרת פתיחת החוליות.

### שימוש באתר 6 דרגות של וויקיפדיה

Six Degrees of Wikipedia[2] הוא פרויקט שמיפה את ערכי וויקיפדיה במטרה למצוא את המספר הנמוך ביותר של מעברים שנדרש כדי לנווט בין כל שני ערכים בוויקיפדיה. הוא מציג בצורה ויזואלית את כל המסלולים הקצרים ביותר בין שני ערכים כאלה. הדאטה בפרויקט, שגם אנחנו עשינו בו שימוש, מגיע מ-Wikimedia, שמייצרת עותק להורדה של ערכי וויקיפדיה באנגלית פעמיים בחודש. ב-Six Degrees of Wikipedia עיבדו את המידע הזה לטבלאות במסד נתונים, ובעזרת שאילתות SQL ביצעו שליפות מהמאגר כדי למצוא את כל המסלולים. האתר היה לנו לעזר, לבדיקות שפיות לאורך תהליך התכנות, לסיעור מוחין בתהליך החשיבה המקדים ולהשוואת מסלולים.

---

[2] אודות אתר שש דרגות של וויקיפדיה: https://www.sixdegreesofwikipedia.com/#
גיטהאב של הפרויקט: https://github.com/jwngr/sdow

## יוריסטיקות

- חיפוש BFS (offline): מימשנו חיפוש BFS קלאסי על מנת להשוות את ביצועי היוריסטיקות לעומת חיפוש סיזיפי, שהולך בדרך שיטתית לעבר מסלול אופטימלי ונותן מושג על רמת הקושי של הבעיה. הוא עוזר גם בניתוח של אורך מסלול שכזה.
- יוריסטיקת הילוך מקרי (offline): נותנת ציון רנדומי לחוליה שהיא מקבלת ובכך מדמה התמודדות ירודה עם בעיה ע"י לחיצה אקראית על לינקים עד לסיום. בעוד שלא הצלחנו לקבל אף לא פתרון אחד למשחק חד-כיווני, משחק דו כיווני הוא פתיר במידה מפתיעה ע"י הילוך מקרי דו-כיווני. רק יוריסטיקות שמצליחות להביס הילוך מקרי ייחשבו בעינינו כראויות.
- יוריסטיקת התפשטות עפ"י לינקים (offline): המטרה היא לנווט לפי מרכזיות הערכים, לנסות לשלוח זרועות לכמה שיותר קליקות ולקוות לפגוש שם את החצי השני של המסלול. על כן, ערך שמגדיל את הפופולריות שלו ביחס לערך הנוכחי יקבל עדיפות גבוהה בפרינג' וערך שלא מקדם את החיפוש או מחזיר אותו אחורה יקבל עדיפות נמוכה. "מרכזיות" נמדדת על-פי כמות הלינקים היוצאים שיש לערך מסוים לעומת הערך שהפנה אליו (המנה בין כמות הלינקים של קודקוד האב לכמות הלינקים של הקודקוד הנוכחי, כאשר ציון נמוך הוא שערוך טוב). בנוסף, קבענו סף לפתיחת קודקודים, והענשנו בשערוך ערכים עם יותר מידי לינקים כדי לא לתקוע את החיפוש. הסף, 800 ללינקים יוצאים ו-500 ללינקים נכנסים, נקבע עפ"י ניסוי וטעייה תוך התבססות על חקירת מסד הנתונים. הערכים שנבחרו לבסוף הם ממוצע הלינקים בחמישה האחוזים של הערכים הכי מקושרים (היוצאים והנכנסים בנפרד).
- יוריסטיקת התפשטות עפ"י שפות (online): יוריסטיקה זו מודדת לכמה שפות ערך מסוים מתורגם לעומת הערך שהפנה אליו. הרעיון מאחוריה הוא שערך משמעותי או מוכר יתורגם לשפות רבות, בעוד ערכים נישתיים או מקומיים יתרגמו פחות. באותו אופן כמו יוריסטיקת הלינקים, ערכים שמשפרים את כמות השפות שהערך שהפנה אליהם מתורגם אליהן ייפתחו קודם. למעשה, יוריסטיקה זו נבנתה כדי להוות השוואה ליוריסטיקת הלינקים ולהפתעתנו היא עקפה אותה בביצועים.

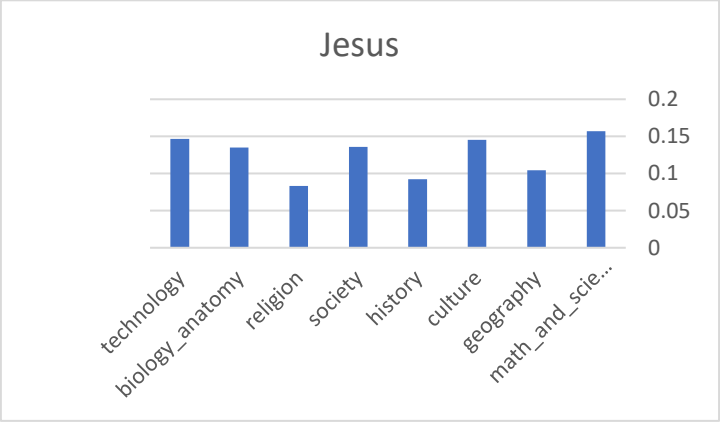
- יוריסטיקת מטא-דאטה (שיטוט ע"פ קטגוריות) (online): ניסיון להביט בשיוך לקטגוריות כסוג של "כיוון" שניתן להחיל על וויקיפדיה. ככל שהערך הנוכחי חולק יותר קטגוריות עם ערך היעד נראה שאנחנו מתקרבים למטרה. על כן היוריסטיקה מעדיפה התקדמות בצירים שחולקים קטגוריות רבות יותר עם היעד. הציון שניתן לכל ערך מחושב ע"י $10^{\text{number of categories shared with goal}}$ . כך אנחנו מזניחים כמעט לחלוטין את אורך המסלול, ומנסים לעלות בהתמדה במספר הקטגוריות שבחיתוך.

### יוריסטיקות המשלבות עיבוד שפה טבעית

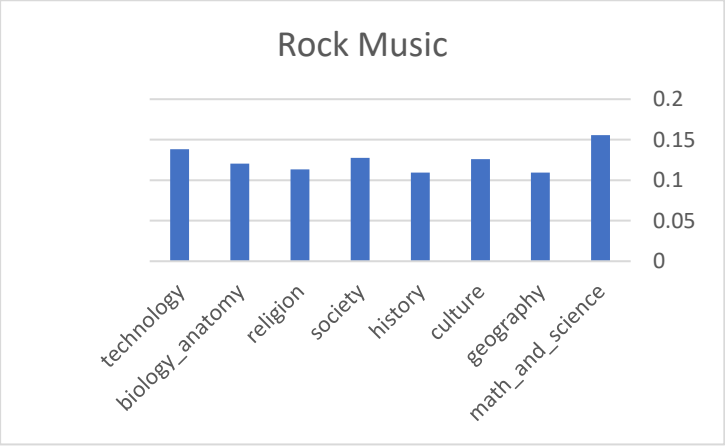
ביקשנו לעבוד עם כלים של למידה ועיבוד שפה טבעית, מתוך ניסיון להתחקות אחר פעולה אנושית.

- יוריסטיקת bag of words (online): יוריסטיקה פשוטה שמבקשת למדוד מרחק בין ערכים לפי השונות במילים שלהן. המרחק בין ערכים מחושב ע"י ספירת מופעים של מילים בכל אחד מהערכים, ייצוגם בוקטור ספירות כ-bag of words ממימד אוצר המילים שנאסף משני הערכים. לבסוף, מדדנו את המרחק האוקלידי בין וקטור הערך הנוכחי ביחס לערך היעד.
- יוריסטיקת פיצ'רים (online): בחרנו לייצג את ערכי הוויקיפדיה במרחב וקטורי, כך שכל קואורדינטה מייצגת פיצ'ר.
  - כל פיצ'ר הוא נושא מבין רשימת הנושאים הבאה: מתמטיקה ומדעים מדויקים, היסטוריה, ביולוגיה ואנטומיה, גיאוגרפיה, תרבות, חברה, דת וטכנולוגיה. כל קבוצה מאופיינת ע"י מספר נציגים, שהם הערכים הבולטים תחת אותו נושא. בחירת הנציגים נעשתה בהתבסס על חלוקה של פורטל וויקיפדיה, תוך איחוד נושאים וסינון חלק מהערכים.
  - כל ערך הוא וקטור, וכל קואורדינטה מכילה ציון שקובע כמה הערך הזה שייך לכל אחד מהנושאים.

חישוב הציון לכל קואורדינטה התבצע בעזרת ייצוג בשיטת bag of words, מדידת מרחק אוקלידי וחישוב ממוצע על פני הנציגים. תחילה ייצרנו אוצר מילים ע"י איסוף כל המילים הייחודיות שהופיעו בכל אחד מהערכים הנציגים בכל אחת מהקבוצות. לכל ערך ייצוג וקטורי כ-bag of words ממימד גודל אוצר המילים, עם ספירת כמות ההופעות של מילה בערך. בנוסף, ייצרנו גם ייצוג וקטורי כ-bag of words לכל אחד מהנציגים. לכל נושא, מדדנו את המרחק האוקלידי בין הייצוג של נציג לבין הייצוג של ערך. הציון הסופי לנושא כולו נקבע לפי ממוצע על ציוני המרחקים מהנציגים. למשל, הערך Jesus מתויג להיות הכי קרוב לדת והיסטוריה, והכי רחוק ממתמטיקה וטכנולוגיה.




ואולם, לא תמיד התיוג היה אפקטיבי, ברוב המקרים הציון על פני הנושאים היה יחסית אחיד




יוריסטיקה זו דורשת מידע רב שיקר היה להשיג: היא מבקשת את הטקסט של הערכים, ובשל מגבלות ה-online שתוארו לעיל, ברוב ההרצות היוריסטיקה לא סיימה לרוץ במסגרת הזמן שהוקצבה לה (חמש דקות).

## הערכת ביצועי היוריסטיקות

כדי ללמוד על ביצועי היוריסטיקות שלנו רצינו לבחון אותן במספר סביבות שונות. ניסינו לבחון האם ליוריסטיקה מסויימת יש יתרון על אחרת כתלות באופי הבעיה שנבחרה. האם יש בעיות מסויימות שבהן היוריסטיקות מצליחות להכות את ה-BFS באופן משמעותי? אולי באחרות לא? כיוון שכל צמד מילים הוא בעיה ייחודית ניסינו להכליל את הבעיות ולחלק אותן לקבוצות באופן שיהיה בעל-משמעות.

### סוגים שונים של בעיות

קיווינו לסווג בעיות למספר רמות קושי. התכנית הייתה לבחור את הערכים הבלתי-קשורים-ביותר שנוכל לחשוב עליהם ולאתגר את האלגוריתם בבעיות נבזיות במיוחד. לאחר מספר נסיונות כאלה גילינו שלא ממש משנה מה היו הערכים שבחרנו, כל צמד ערכים שהצלחנו להעלות בדעתנו התבררו כמקושרים דרך 2-4 דילוגים. (למשל: סקוטלנד יארד ← לונדון ← דינוזאור). BFS התמודד איתם בגבורה, ולא נדרש שימוש ביוריסטיקות. שייכנו את התצפית הזו לתופעת "הטיית הבחירה", הטייה קוגנטיבית המאפיינת נטייה של חוקרים להאמין שהערכים שנדגמו על ידם בניסוי משקפים במידה טובה את העולם וגורמת להם להזניח את העיסוק באופן הדגימה שאולי משפיע מאד על התוצאה.

כך קבענו את קבוצת הבעיות הראשונה: **הבעיות האימפולסיביות**, המאופיינות בערכי התחלה וסוף שהוגרלו באופן אחיד מתוך רשימת 5,000 הערכים הפופולריים ביותר (הכי הרבה כניסות) המתעדכנת בתדירות שבועית[3].

כקונטרה מתבקשת לבחירת ערכים פופולריים הגדרנו את קבוצת הבעיות השניה להיות קבוצת **הבעיות הרנדומיות** בה בוחרים את ערך המקור והמטרה בעזרת מנגנון "שליפת ערך אקראי" שמספקת וויקיפדיה.

לאחר מכן ניסינו לאפיין צמדים קשים במיוחד. לשם כך נעזרנו בDB ה-Offline וחיפשנו ערכים נידחים במיוחד, כלומר כאלה שמעט לינקים מפנים אליהם והם מפנים למעט ערכים אחרים, אלה

[3] חמשת אלפים העמודים הנצפים ביותר בוויקיפדיה בכל שבוע:
https://en.wikipedia.org/wiki/User:West.andrew.g/Popular_pages

יכונו ערכים נישתיים. ב**בעיות הנישתיות** מגרילים באופן אחיד ערך התחלה וסוף שמקיימים את תכונת הנישתיות. כמו כן אפיינו ערכים שנכנה splitters, כאלה שמפנים להרבה ערכים אבל מעט מפנים אליהם, ו-mergers, כאלה שמוצבעים ע"י הרבה ערכים אבל מפנים בעצמם למעט. ב**בעיות טבעיות** מגרילים ערך התחלה splitter וערך סיום merger, וב**בעיות אקסטרים** מגרילים ערך התחלה merger וערך סיום splitter.

הנחנו שמרחב הפתרונות יהיה עשיר ורווי במסלולים מוצלחים בבעיות טבעיות, יותר מאשר בבעיות אקסטרים. על כן, ציפינו שליוריסטיקות יהיה קל יותר לפתור בעיות טבעיות ולהגיע לתוצאה טובה. גם אם הייתה הערכה שגויה בדרך ששלחה את החיפוש לאיזור מרוחק מאחד הפתרונות בגרף, יהיו מספיק מסלולים חלופיים מוצלחים שהאלגוריתם יוכל למצוא.

## איך מודדים?

כדי להעריך את מידת ההצלחה של היוריסטיקות בבעיות השונות ויתרנו על דרישת המהירות באופן חלקי. חלק מהיוריסטיקות עובדות Online, כך שזמן משיכת המידע הוא הפקטור המשמעותי ביותר בריצת האלגוריתם, וחלקן עובדות Offline. לכן, אין ערך של ממש בהשוואת זמני הביצוע. השארנו הגבלת timeout ועצרנו משחקים שלא נפתרו תוך חמש דקות, זמן סביר שבמהלכו אדם יכול למצוא מסלול, ואנחנו מעוניינים ביוריסטיקות שיוכלו לגבור על שחקנים אנושיים.

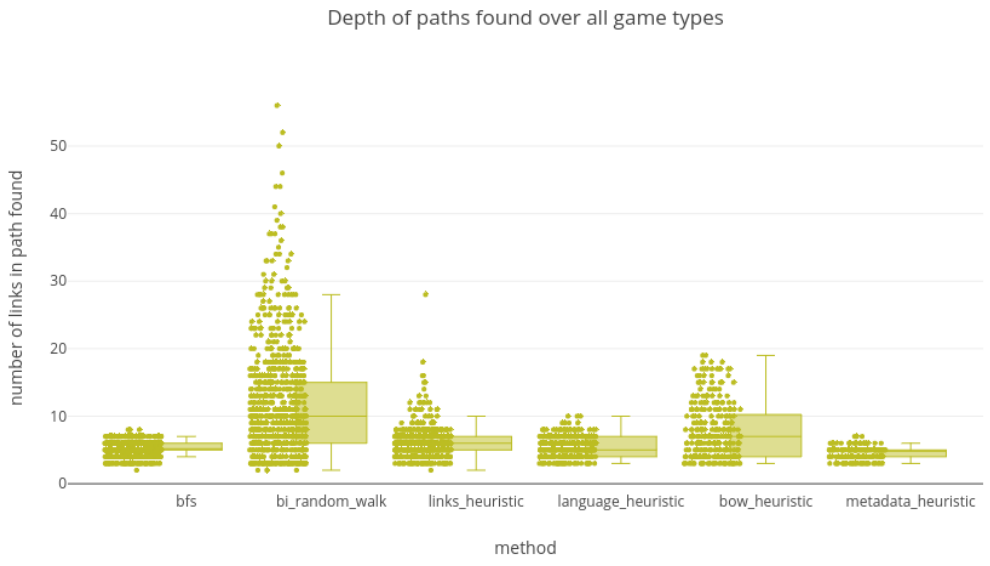
החלטנו להעדיף את מדד כמות החוליות שנפתחו על פני מספר הדילוגים במסלול, כיוון שלא נדרש שימוש בבינה מלאכותית כדי למצוא מסלולים קצרים, BFS מצליח גם בלי להבין כלום. לעומת זאת, פתיחה של מספר קטן של חוליות היא מאפיין של שחקן תבוני שעושה פעולות מכוונות כדי לסיים את המשחק במהירות ובהצלחה. על כן זהו המדד העיקרי המשמש אותנו להערכת ביצועי היוריסטיקות השונות.

## תוצאות ודיון

כדי לאמוד את ביצועי היוריסטיקות השונות הקמנו מכונה וירטואלית על Google Cloud Platform והטלנו עליה להריץ מאות בעיות מכל סוג ולפתור אותן תוך שימוש ביוריסטיקות השונות במשך שבוע. לוג התוצאות זמין וניתן לעיין בו כדי לחקור מקרוב את המשחקים השונים ששוחקו[4].

נביט תחילה על אורכי המסלולים האופייניים שהיוריסטיקות מוצאות. בגרף הבא כל נקודה מייצגת משחק נפרד. מיקומו על ציר האיקס מציין את היוריסטיקה, וגובה הנקודה מייצג את אורך המסלול שנמצא. Boxplot מתאר את פיזור התוצאות האופייניות לכל יוריסטיקה בנפרד. משחקים שלא צלחו אינם מסומנים בגרף, וניתן להסיק ממנו רק את גודל המסלולים האופייניים שהוא נוטה להפיק.

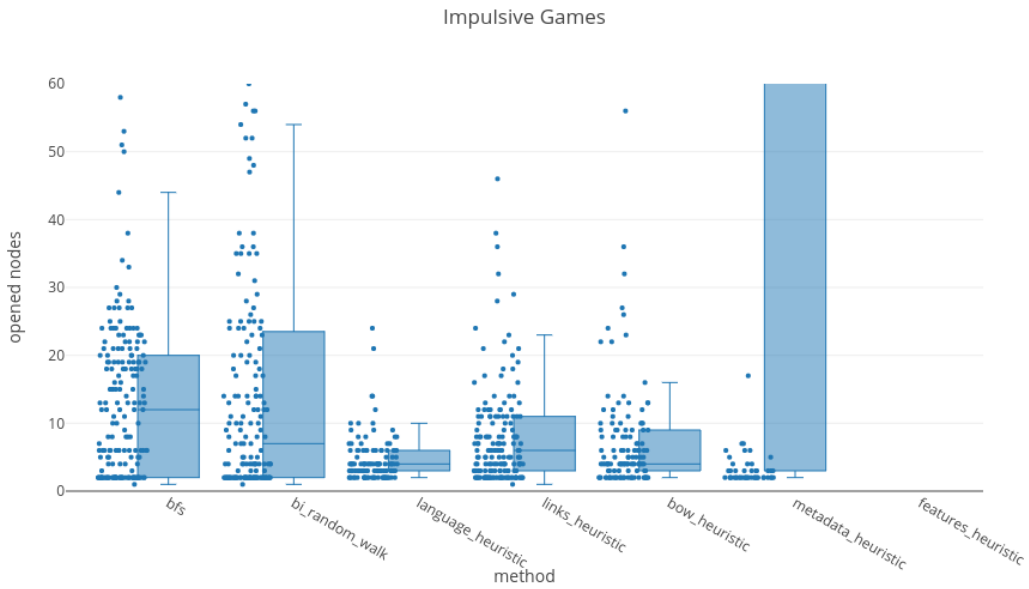
BFS מפיק את אורכי המסלולים האופטימליים, לכן יוריסטיקות שמראות טווח אורכי מסלולים דומה הן מעשיות. באסטרטגיית ההילוך המקרי, החיפוש בוחר באקראי לינק ולוחץ עליו. יוריסטיקות שמוצאות מסלולים ארוכים יותר לא מצליחות לגבור אפילו על אסטרטגיה טריוויאלית.

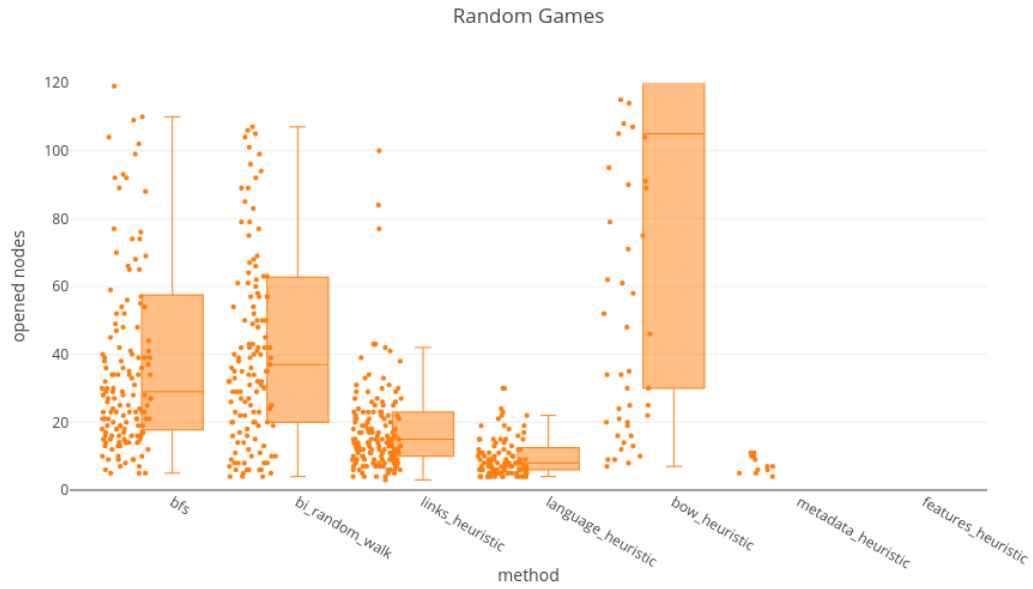



לשמחתנו, היוריסטיקות השונות מצאו מסלולים טובים באופן מובהק מיוריסטיקת ההילוך המקרי, ויוריסטיקות חיפוש דו-כיווני סטטי אוניברסלי נטו לביצועים שדומים ל-BFS מבחינת אורך מסלול.

[4] תוצאות הרצת היוריסטקיות: https://plot.ly/~MitzitTheCat/8

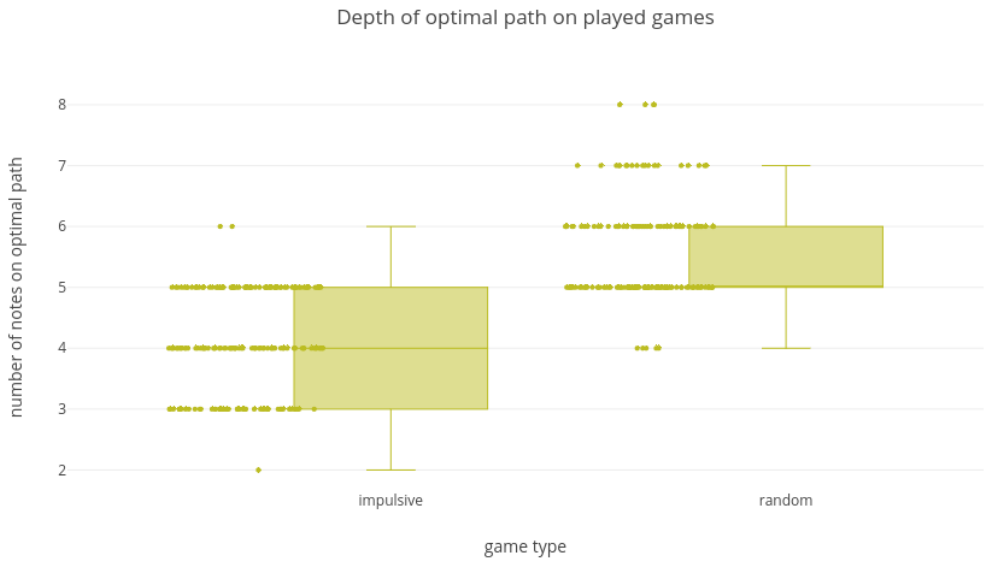
הגרפים הבאים מתארים את ביצועי היוריסטיקות השונות עבור כל קבוצת בעיות בנפרד. כל נקודה מייצגת משחק, גובהה מייצג את מספר החוליות שנפתחו. לצידן מוצג Boxplot שמתאר בצורה מסכמת את החציון ואת התוצאות האופיניות לכל שיטה. השמטנו רשומות שנכשלו בגלל שגיאות טכניות ונתנו לכל המשחקים שנכשלו ב-timeout ערך פיקטיבי של 2000 חוליות שנפתחו. ערך זה הוא גבוה באופן מובהק מכל המשחקים התקניים וקל לזהות אותו (נבחין שלא כל משחק הוא פתיר). נשווה כעת בין תוצאות המשחקים האימפולסיביים והמשחקים הרנדומיים.



אפשר לזהות שטכניקות ה-BFS וההילוך המקרי פתרו משחקים מקטגוריית הבעיות האימפולסיביות בעזרת פתיחה של הרבה פחות חוליות מאשר במשחקים מקטגוריית הבעיות הרנדומיות. לכן, אנחנו משערים שמשחקים אימפולסיביים הם באופן אינהרנטי קלים יותר, ושמרחק בין ערכים פופולריים הוא קטן מהמרחק הכללי בין צמד ערכים.

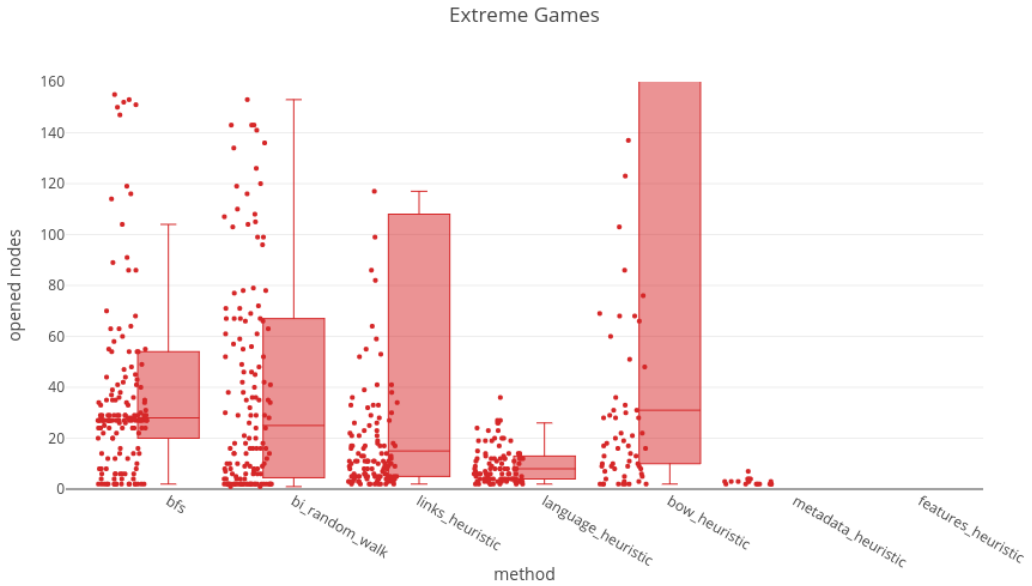
השערה זו מקבלת גיבוי מהנתונים, שמראים שמסלולים אופטימליים המאפיינים משחקים אימפולסיביים הם באורך 4 ומסלולים אופייניים במשחקים רנדומיים הם מאורך 5. למעשה מסלולים מאורך 4 הם זוג מסלולים מאורך 2, כלומר דורשים מעבר על פני פיצול אחד של העץ מכל צד של ריצת האלגוריתם הדו-כיווני. לעומתם מסלולים מאורך 5 דורשים לפחות צד אחד מאורך 3. לכן, משחקים באורך 4 הם רדודים בסדר גודל מעריכי (דורשים פתיחת רמה אחת פחות בעץ) מאשר משחקים עם מסלולים מאורך 5.

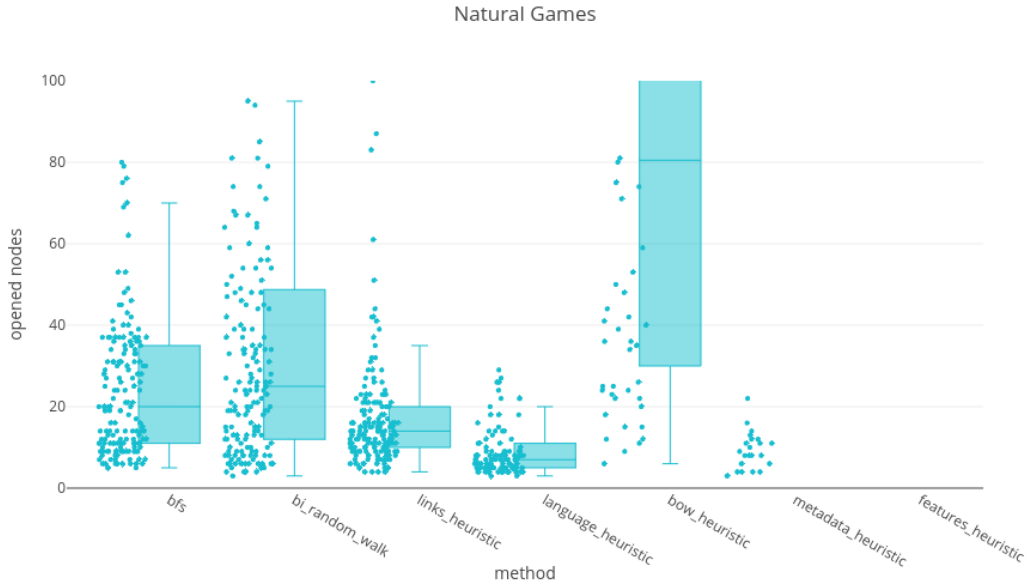



הבחנה זו היא חשובה במיוחד כשמנסים להסביר את השוני הגדול בביצועי יוריסטיקות bag of words ומטא-דאטה (יוריסטיקת הקטגוריות) בין בעיות אימפולסיביות לבין בעיות רנדומיות. הסבר מניח את הדעת לפער הוא שבעיות אימפולסיביות באמת אינן קשות מספיק כדי להשתמש בהן כבחינה אובייקטיבית ליוריסטיקות, ואפילו היוריסטיקות החלשות מצליחות בהן היטב. הגיוני גם שבעיות אימפולסיביות הן עשירות במסלולים קצרים, ועל כך תעיד העובדה שההילוך המקרי נמוך באופן משמעותי מאשר ב-BFS.

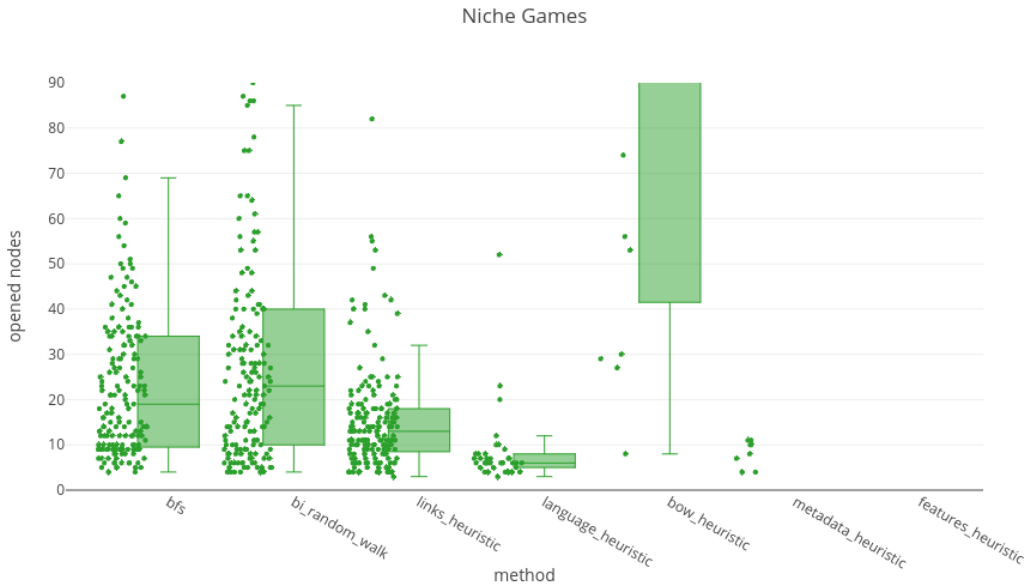
נתמקד אם כן בניתוח המשחקים הרנדומיים. ניכר שאסטרטגיות דו-כיווניות אוניברסליות שולטות במשחק, והדו-כיווניות שמניחות מונוטוניות במרחב הוויקיפדיה מפגינות ביצועים מאכזבים. נבחין גם שיוריסטיקת פיצ'רים לא הצליחה לפתור אפילו בעיות אימפולסיביות.

נביט בתוצאות שהתקבלו עבור בעיות אקסטרים ובעיות טבעיות:



החלוקה הברורה בין אסטרטגיות אוניברסליות למונוטוניות נשמרת כאן ומקבלת משנה תוקף. יוריסטיקות שמניחות מונוטוניות לא מצליחות להתמודד עם האתגר בלפחות חצי מהמקרים. ניכר גם שמשחקי האקסטרים הצליחו לבלבל ולהטעות את יוריסטיקת הקישורים במידה רבה יותר מאשר את יוריסטיקת השפות. אנחנו מניחים שאם היינו מבצעים ניסוי דומה שכולל בעיות שנבחרו על סמך מניפולציה כלשהי על כמות השפות שאליהן ערך מתורגם היינו מקבלים תוצאות הפוכות בהתאמה. נביט בתוצאות המשחקים הנישתיים:




משחקי הנישה מספקים אישוש נוסף למסקנה העיקרית בדבר עליונות היוריסטיקות האוניברסליות על המונוטוניות ומצביעים על יתרון מובהק ליוריסטיקת שפה על פני יוריסטיקת לינקים.

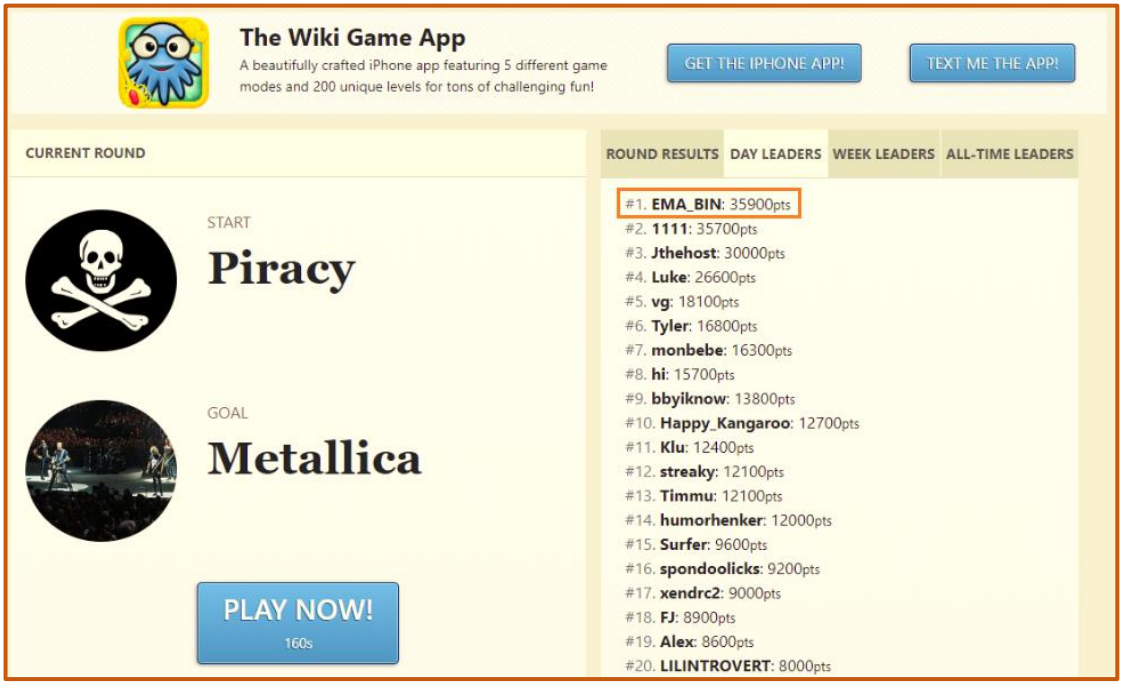
## מסקנות

נסיק שמספר השפות שאליו תורגם ערך מהווה אינדיקציה טובה יותר מאשר כמות לינקים לאפיון מידת החיוניות או המרכזיות של ערך במרחב הוויקיפדיה. כמו כן, ניסיונות בסיסיים לשכן את וויקיפדיה במרחב אוקלידי נכשלו. ייתכן שהמימוש האינטואיטיבי שהצענו לא מספיק מתוחכם מכדי לתת הטלה משמעותית למרחב וקטורי. יחד עם זאת, לא הצלחנו לייצר פתרון מספיק "רזה" מבחינת כמות הבקשות לוויקיפדיה וחסכון בזמני עיבוד. ייתכן שבעזרת כלים רבי עצמה ניתן להכפיף את וויקיפדיה לקירוב של מרחב וקטורי. לעת עתה, הכלים הטובים ביותר שייצרנו מנסים להתחקות אחר המבנה הטבעי המורכב והמרתק של מרחב זה, ומצליחים לא רע במשימה.

## מדידה אקסטרינזית של הפתרון (יוריסטקת לינקים)

החלטנו לבדוק את איכות הפתרון גם מול שחקנים אחרים ברשת. אחרי שראינו שיוריסטיקת הלינקים ה-Offline עובדת מספיק מהר, מימשנו את הסקריפט OnlineGamer.py. הסקריפט המבוסס selenium (ספריית אוטומציה ב-Python) שיחק באתר thewikigame.com, אתר שבו משחקים גולשים רבים במשחק הוויקיפדיה בממשק נוח ומזמין, בו השתמשנו כזירה לבחון את הצלחת יוריסטיקת הלינקים האוניברסלית מול יריבים אמיתיים. כל 300 שניות מתחלפים ערכי התחלה וסיום והשחקנים מקבלים נקודות על מציאת מסלולים קצרים.

הסקריפט שיחק תחת השם EMA_BIN (ראשי תיבות של אלה, מאיה, אלון בשילוב קיצור של בינה מלאכותית). תוך כמה שעות EMA כבשה את ראש הטבלה היומית. כלומר, הפתרון שלנו טוב גם ביחס לשחקנים אנושיים שמנסים לפתור את הבעיה במשימה מהעולם האמיתי.




חושבים שאתם חכמים יותר מEMA? אנחנו מזמינים אתכם להוריד את הפרוייקט ולהתחרות ראש בראש עם האלופה.

## התקנה והרצת הפרויקט

כדי להריץ את הפרויקט באמצעות Python 3.5 יש לוודא כי החבילות הבאות מותקנות:

- wikipedia
- sqlite3
- BeautifulSoup
- requests
- sklearn
- scipy
- selenium (עבור ה-OnlineGamer)

כדי להריץ את היוריסטיקות הOffline יש להוריד את קובץ מסד-הנתונים ולחלץ אותו עם השם "sdow.sqlite" בצמוד לתיקיית EMAbin (לתיקייה שבתוכה נמצא הפרויקט, ולא לתוך תיקיית הפרויקט).

קובץ ההרצה הוא הexecutor.py, והרצתו ללא ארגומנטים תדפיס הודעת usage. בהנתן שמות יוריסטיקות לא תקינים, יודפסו שמות היוריסטיקות האפשריות.

בשביל לראות את הקוד מתחרה באתר https://thewikigame.com יש להריץ את הקובץ OnlineGamer.py ללא ארגומנטים.