תרגיל מס׳ 2: פתרון בעיית N-Puzzle בעזרת אלגוריתם A\*

**מגישים: אביאל ברקוביץ' (ת"ז: 211981105) ומאיר קלמפנר (ת"ז: 211954185)**

עליכם לעבור על הקבצים המצורפים, להבין אותם ולבצע את המשימות הבאות:

1. שנו את הפונקציה ההיריסטית כך שתחזיר את מספר האריחים שאינם במקומם.

def hdistance1(s):        # for h1 (tiles out of place)

    n = len(s[0])

    out = 0

    for i in range(n):

        if s[0][i] != i:  # checks if the piece in the board is in place

            out += 1      # if the piece is not in place, we count it

    return out

1. הריצו את search(3) ו-search(4) עם היריסטיקה, וכתוב תוכנה שיפלוט את העומק המקסימאלי ואת מספר הבדיקות (items) בממוצע מ100 ריצות מחיפושים בפאזלים של 3X3 ו4X4. הפלט יכול להיות עבור search(3):

Average depth 1.83

Average items 6.39

def avg\_search(times, n):

    sum\_depth = 0

    sum\_items = 0

    for i in range(times):

        \_, items, depth = search(n)

        sum\_depth += depth

        sum\_items += items

    avg\_depth = sum\_depth / times

    avg\_items = sum\_items / times

    return avg\_depth, avg\_items

print("--- Starting average search of 3 ---")

depth3, items3 = avg\_search(100, 3)

print("Average depth", depth3)

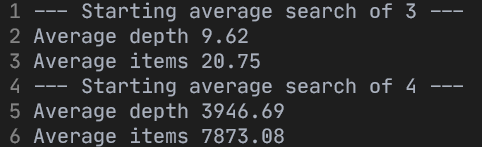
print("Average items", items3)

print("--- Starting average search of 4 ---")

depth4, items4 = avg\_search(100, 4)

print("Average depth", depth4)

print("Average items", items4)



1. שנו את הפונקציה ההיריסטית כך שתחזיר את סכום מרחקי מנהטן של האריחים ממקומם במצב המטרה.  
     
   כדי לבדוק ששני הפונקציות עובדות, הריצו כל אחד היוריסטיקות עבור הלוח הבא במקום לוח רנדומאלי:

def hdistance2(s):                   # for h2 (Manhattan)

    n = int(len(s[0])\*\*0.5)

    distance = 0

    for i in range(n\*\*2):

        target\_row = i // n

        target\_col = i % n

        row = s[0][i] // n

        col = s[0][i] % n

# calculates the differences between the rows and the columns

        distance += abs(target\_row - row) + abs(target\_col - col)

    return distance

s=[[4, 3, 7, 5, 8, 6, 1, 0, 2], '']

מה הפלט בשביל שני היריסטיקות?

יוריסטיקה מספר 1:



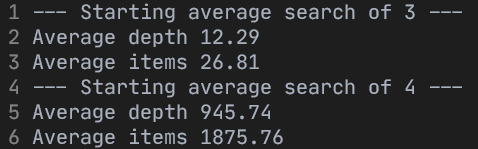
יוריסטיקה מספר 2:



1. הריצו את search(3) ו-search(4) עם היריסטיקה, וכתוב תוכנה שיפלוט את העומק המקסימאלי ואת מספר הבדיקות (items) בממוצע מ100 ריצות מחיפושים בפאזלים של 3X3 ו4X4. הפלט יכול להיות עבור search(3):

Average depth 1.83

Average items 6.39



**מה מסקנתכם?**

מניתוח התוצאות דלעיל עולה כי היוריסטיקת h2 המבוססת על חישוב מרחקי מנהטן יעילה בהרבה מהיוריסטיקת h1 המבוססת על חישוב מספר החלקים שאינם במקום.