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

מצורפים הקבצים הבאים:

search.py – מממש את אלגוריתם החיפוש.

frontier.py – מממש ערמה.

state.py – מממש מצב של N-Puzzle בגודל nXn. הפונקציה ההיריסטית מחזירה 0 לכל מצב.

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

1. שנו את הפונקציה ההיריסטית כך שתחזיר את מספר האריחים שאינם במקומם.
2. 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

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:

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

מה הפלט בשביל שני היריסטיקות?

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?**