

מבוא לבינה מלאכותית

תרגיל בית 1 – תשפ"ב

סיכום ממצאים מ-הקוד

(1) תיאור ההיוריסטיקה: בחרנו את היוריסטיקה להיות מספר האיברים שלא נמצאים במקומם במטריצה וזה אומדן שעומד בהגדרה של היוריסטיקה קבילה מכיוון ש לא ייתכן שהמסלול ליעד יהיה קטן ממספר האיברים שלא נמצאים במקומם כי הרי במקרה הטוב אנחנו נעשה X החלפות כדי להגיע ליעד (X הוא מספר האיברים שלא נמצאים במקומם) לכן מסיבה זו היוריסטיקה הינה היוריסטיקה קבילה.

(2) ביצועים של האלגוריתמים השונים על הקלטים שלנו:

	inpMatOne. txt	inpMatTwo. txt	inpMatThree. txt	inpMatFour. txt	Cour se input txt
A*					
Nodes visited	6	3	5	5	63
Steps to goal	4	2	4	4	13
bfs					
Nodes visited	1056	52	8308	14042	3202 1
Steps to goal	4	2	4	4	13
Iterative deepenin g					
Nodes visited	697	34	5384	9626	1498 2
Steps to goal	4	2	4	4	13

(3) מסקנות לגבי האלגוריתמים השונים:

- ניתן לראות לפי הטבלה ולפי החומר הנלמד בכיתה, A^* הינו האלגוריתם המוצא את הפתרון בכמות צעדים הכי נמוכה והוא אכן מוצא את הפתרון באופן אופטימלי, אלגוריתם BFS נתן ביצועים לא טובים במקרה שלנו כי הפתרון נמצא בשכבה מאוד עמוקה והוא מחפש לפי שכבות, ID עבד יותר טוב מ BFS אבל ההבדל שהוא הולך לעומק ובגלל שבעומק היו פתרונות אז ה ID נתן ביצועים יותר טובים מ BFS

- המסלולים ליעד כולם שווים לכן המסקנות לגיטימיות לגבי האלגוריתמים השונים.

- אז לסיכום האלגוריתם המועדף ביותר במקרה שלנו הינו A^* ואחריו בא DFS בכמה מקרים BFS שניהם כמעט דומים והשונה ביניהם נגרם מהקלט שניתן לתוכנית

: **CMD**

```
rc>javac SolveCl.java
\EX1ai\src>java SolveCl
Solve: inpMatOne.txt bfs
```

שם הקובץ טקסט

שם האלגוריתם
שתרצה להפעיל

התממשקות עם ה
JAVA

Command Prompt

```
javac StateCl.java
javac SolveCl.java
java SolveCl
olve: inpMatTwo.txt bfs
3 2
6 5
8 9
=====
1 3 2
6 5
8 9
=====
3 2
6 5
```