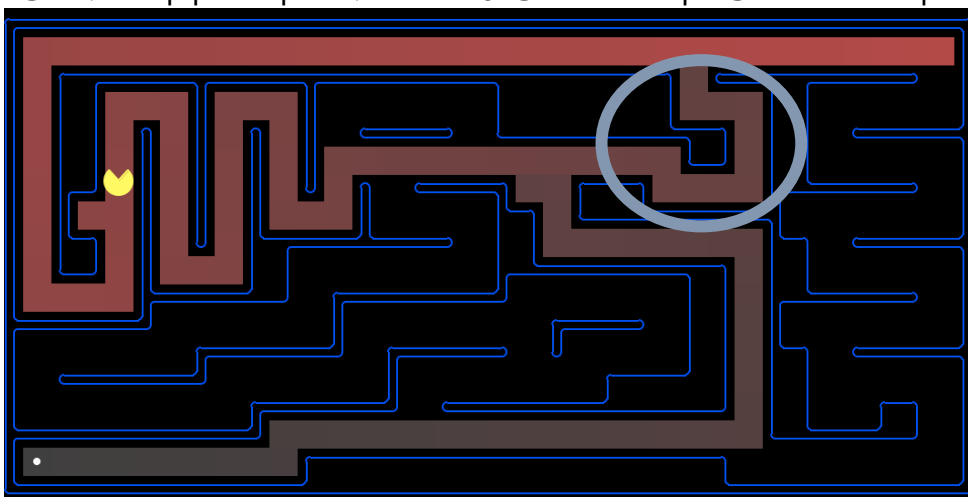
דו"ח תרגיל 1- בינה מלאכותית

1.1) האם סדר החיפושים הוא מה שהיית מצפה? האם פאקמן באמת הולך לכל הצמתים שנחקרו בדרך לנקודת הסיום? (ציין תשובותך בדו"ח)

רמז: אם אתה משתמש ב- Stack כמבנה הנתונים שלך, הפתרון שנמצא על ידי אלגוריתם DFS שלך עבור mediumMaze צריך להיות באורך של 130 (בתנאי שתכניס את הבנים בסדר בו הם התקבלו ב getSuccessors; ייתכן שתקבל 246 אם תכניס אותם בסדר ההפוך).

1.2) האם זהו פתרון בעלות נמוכה ביותר? אם לא, חשוב מה החיפוש הראשון DFS עושה לא בסדר. (ציין תשובתך בדו"ח)

1.1) סדר החיפושים באלגוריתם הDFS הוא מה שהייתי מצפה- פאקמן הולך לעומק במסלול אחד ומגיע למטרה ולמרות שהוא לא סרק את כל הצמתים האפשריים, התוכנית מסתיימת. בנוסף פאקמן לא הולך לכל הצמתים שנחקרו- הצמתים שמסומנים בעיגול נחקרו ופאקמן לא הגיע לשם.



1.2) DFS אינו אופטימלי, ובפרט במבוך הנ"ל פאקמן אינו בוחר במסלול הקצר ביותר במבוך. מה שהאלגוריתם עושה הוא ראשית ללכת לעומק ולמצוא אם במסלול ספציפי ניתן להגיע לפתרון. אם הוא מוצא פתרון אז התוכנית מסתיימת, גם אם יכול להיות מסלול יותר אופטימלי ברמה יותר גבוהה. את רמת הביצועים הזאת ניתן לקבל בסריקה לרוחב.

4.1) מה קורה ב openMaze באסטרטגיות החיפוש השונות?

4.1) ההרצה ראשונה- dfs, הרצה שניה- bfs, הרצה שלישית- ucs, הרצה רביעית- astar

```
Daniels-MacBook-Air-4:ex1 Daniel$ python3 pacman.py -l openMaze -z .5 -p SearchAgent -a fn=dfs,heuristic=manhattanHeuristic
[SearchAgent] using function dfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 298 in 0.1 seconds
Search nodes expanded: 576
Pacman emerges victorious! Score: 212
Average Score: 212.0
Scores:        212.0
Win Rate:      1/1 (1.00)
Record:        Win
Daniels-MacBook-Air-4:ex1 Daniel$ python3 pacman.py -l openMaze -z .5 -p SearchAgent -a fn=bfs,heuristic=manhattanHeuristic
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.1 seconds
Search nodes expanded: 682
Pacman emerges victorious! Score: 456
Average Score: 456.0
Scores:        456.0
Win Rate:      1/1 (1.00)
Record:        Win
Daniels-MacBook-Air-4:ex1 Daniel$ python3 pacman.py -l openMaze -z .5 -p SearchAgent -a fn=ucs,heuristic=manhattanHeuristic
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.1 seconds
Search nodes expanded: 682
Pacman emerges victorious! Score: 456
Average Score: 456.0
Scores:        456.0
Win Rate:      1/1 (1.00)
Record:        Win
Daniels-MacBook-Air-4:ex1 Daniel$ python3 pacman.py -l openMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
[SearchAgent] using function astar and heuristic manhattanHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.1 seconds
Search nodes expanded: 535
Pacman emerges victorious! Score: 456
```

כפי שניתן לראות, באלגוריתם dfs מפותחות 576 צמתים, בbfs וucs 682 כל אחד, ובastar 535 צמתים.
למרות זאת, העלות בdfs גדולה בפער משאר האלגוריתמים וזאת מכיוון שאנו בוחרים מסלול שאינו אופטימלי.
בשלושת האלגוריתמים האחרים יש אופטימליות ולכן העלות גם זהה.

5.1) תאר בדו"ח את המצב שבחרת לייצג.

5.1) המצב שבחרתי לייצג הוא זוג (tuple) של המיקום של פאקמן ורשימה של כל הפינות שטרם ביקרנו בהן.
בחרתי לייצג את המצב כך כיוון שזהו המידע הבלעדי שרלוונטי לבעיה. בעזרת המצב הזה אני יכול לדעת מתי הגעתי למטרה שלי כי יש לי את המיקום שלי שבעזרתו אני יודע אם הגעתי לפינה וכתוצאה מכך אם הרשימה שלי ריקה.

6.1) תאר הפו' היוריסטית שבחרת, והסבר את נכונותה.

6.1) הפונקציה היוריסטית שבחרתי היא "בחר את המרחק (מרחק מנהטן) לפינה שהכי רחוקה ממך".

נכונות:
הפונקציה קבילה- הפונקציה שבחרנו תמיד תהיה זולה או שווה לעלות הזולה ביותר, מכיוון שלא ייתכן שהעלות האמתית יותר זולה מהמרחק לפינה הרחוקה ביותר שעלולה גם לרוקן את רשימת הפינות שטרם ביקרנו בהן ולהיות מצב המטרה. בנוסף מרחק המנהטן מתעלם מקירות המבוך כך שלא יכול להיות שפאקמן יגיע לפינה הרחוקה ביותר תוך מספר צעדים נמוך יותר.

הפונקציה עקבית- הפונקציה חיובית (עלות צעד ממצב למצב הוא לפחות 1) ולכן מתקיים
$h(n) \leq c(n, a, n') + h(n') \leq 1 + h(n)$
לכן הפונקציה עקבית.

7.1) תאר הפו' היוריסטית שבחרת, והסבר את נכונותה.

7.1) הפונקציה שבחרתי היא "בחר את המרחק (מרחק בbfs) לאוכל שהכי רחוק ממך".

נכונות:
הפונקציה קבילה- הפונקציה שבחרנו תמיד תהיה זולה או שווה לעלות הזולה ביותר, מכיוון שלא ייתכן שהעלות האמתית יותר זולה מהמרחק לאוכל הרחוק ביותר שעלולה גם להיות האוכל האחרון שבלוח ולרוקן את foodList וכתוצאה מכך להיות מצב המטרה. אחרת, נסתור את אופטימליות bfs בחישוב מסלול קצר ביותר ממצב למצב.

הפונקציה עקבית- הפונקציה חיובית (עלות צעד ממצב למצב הוא לפחות 1) ולכן מתקיים
$h(n) \leq c(n, a, n') + h(n') \leq 1 + h(n)$
לכן הפונקציה עקבית.