

בס"ד

פרויקט בבינה מלאכותית

משחק הלוח מאסטר-מיינד



מגישות: מוריאן פרץ, הודיה מושייב, קציעה יצחק

פרויקט בינה מלאכותית - משחק הלוח מאסטר-מיינד

1. מבוא הסבר המשחק ומטרתו

מדובר במשחק ל 2 שחקנים בו על כל שחקן לפצח את קוד היריב, כאשר בתחילת המשחק כל שחקן בוחר קוד משלו, הקוד הינו בעל 4 ספרות יכול להכיל 6 צבעים כל שחקן מוגבל ל 12 נסיונות של ניחוש על כל ניחוש מקבל השחקן מהיריב ניקוד על הניחוש

- נקודה שחורה אם המקום והצבע תואם לקוד היריב
- נקודה לבנה אם רק הצבע תואם
- אין נקודות במידה ואין שום התאמה

ניתן לבחור את אותו צבע יותר מפעם אחת, הניקוד ניתן בכללי על הניחוש ולא על כל נקודה בניחוש ככה שאי אפשר לדעת איזה נקודה מתוך הבחירה הייתה נכונה

מטרת המשחק לאפשר לשחקן למצוא במינימום ניחושים מה הקוד של היריב.

בחרנו להציג פתרון לבעייה פרטית בגודל קוד 4 ו 6 צבעים

2. הסבר/תיאור של האלגוריתם:

האלגוריתם יחזיר לכל ניחוש את האופציות שיכולות להיות התשובה הנכונה, כך שמצד אחד מספר האופציות יהיה הגדול ביותר (כי אז הסיכוי יותר גדול ששם תהיה התשובה הנכונה), ומצד שני לקחת את הקטן ביותר מבין כל המקסימליים כדי לצמצם את מספר הניחושים במהלכים הבאים.

להלן הסבר עבור ספירת האפשרויות עבור כל ניקוד אפשרי- כדי להבין את תהליך הספירה ננסה להבין מהלך ראשון במשחק שלהלן: ניקח מרחב קטן יותר של אפשרויות לדוגמא: בחירת 2 צבעים מתוך 3 (א-אדום, כ - כחול, צ- צהוב)

מספר הסיידורים האפשריים הוא 3^2 שזה תשע אפשרויות ((א, א), (א, כ), (א, צ), (כ, כ), (כ, צ), (צ, צ), (צ, כ), (צ, א), (כ, א))

מספר התוצאות האפשריות הן 5 (פ,פ),(פ,ב),(ב,פ),(ב,ב), (,)

עבור המהלך הראשון נבנה טבלה שמציגה לי כמה תוצאות אפשריות נמצאות לי בתוך כל בחירה שאתן - משנה סוג הבחירה ולכן נבחר xx או xy

xx	xy	
1	1	בול, בול
0	1	פגיעה, פגיעה
4	4	בול
0	2	פגיעה
4	1	כלום

לפי הטבלה אם ננחש שני צבעים זהים השחקן מולנו יעניק לנו אחד משלושת הציונים הבאים (בול, בול)-כשקלענו, (בול,) = הסיכוי שיתן לנו בול אחד הוא פעמים עבור כל רצף שמחזיק את הצבע (,)-כשאין התאמה באופן דומה העמודה השנייה כוללת ניתוח של התפלגות הרצפים החבויים האפשריים בהתאם לציון שנקבל אם ננחש ניחשו שבו שני צבעים שונים באלגוריתם שמימשנו נבחר את סוג הניחוש שהערך המקסימלי בעמודה שלו הוא הקטן ביותר בדוגמא לעיל אין הבדל מה בוחרים כי הספרה 4 מקסימלית בשניהם

מגבלות האלגוריתם:

מדובר בבעיית np-complete במידה ומגדילים את הלוח וכן את מספר הצבעים לבחירה הבעיה גדלה אקספוננציאלית, לכן כדי שהאלגוריתם יעבוד וירוך בזמן סביר, נאלצנו לצמצם את מרחבי הבעיה ל6 מספרים בלבד ו4 מקומות אפשריים לכל מספר. בנוסף, אם האלגוריתם לא מפצח את הקוד תוך מספר מוגבל של ניחושים, הוא נעצר ולא ממשיך לחפש.

3. שיטות: השיטה שבה עובד האלגוריתם הוא מעבר על כל האפשרויות הקיימות ובחירת הניחוש האופטימלי בכל שלב, לאחר הורדת האפשרויות שבטוח אין הפתרון על סמך הניחוש הקודם, היתרון באלגוריתם שהוא מביא פתרון בתוך מספר שלבים מצומצם עד 5 שלבים

הפונקציות הקיימות:

evaluate

מקבלת: ניחוש ותשובה

ומחזירה: מספר הנקודות הלבנות והשחורות וריק במידה ולא קיים

random_list

מקבלת: גודל קוד ומספר צבעים לבחירה

ומחזירה: קוד רנדומלי בגודל הנדרש מתוך הצבעים לבחירה

possible_answers

מקבלת: גודל קוד ומספר צבעים לבחירה

ומחזירה: את כל הצירופים האפשריים במרחב כלומר עבור קוד בגודל 4 עם 6

אפשרויות בחירה יחזור לי 1296 צירופים אפשריים 6^4

check_winner

מקבלת: ניחוש ותשובה

ומחזירה: אם זיהנו את הקוד או לא

heuristic

מקבלת: כל הסידורים האפשריים, ניחוש אחרון, ניקוד עבור הניחוש, קבוצת

הסידורים האפשריים לאחר סינון

מהלך הפונקציה-

- תחילה עוברים על קבוצת הניחושים האפשריים לאחר סינון, ומורידים מהקבוצה את כל הערכים שהניקוד שלהם שונה מהניקוד שהתקבל עבור הניחוש הראשון
- עוברים על כל הניחושים האפשריים ויוצרים עבור כל ניחוש ניתוח של התפלגות הרצפים החבויים האפשריים, בהתאם לציון שנקבל אם ננחש ערך עבור הקבוצה שמופיעה בנקודה הראשונה שלעיל
- נקבל את הערך הגבוה ביותר מהקבוצה ונשמור במערך, נעבור על הערכים בקבוצה ונקבל את הערך הנמוך ביותר שהתקבל
- נחזיר את הערך הנמוך ביותר בתור הניחוש הבא האופטימלי

מחזירה את הניחוש הבא האופטימלי וכן את הקבוצה החדשה לאחר הורדה של כלל הסידורים האפשריים לפי המידע שהתקבל בניקוד האחרון

4. אימות התוכנית:

הרצנו את התוכנית עם 6^4 עבור 20 משחקים וכן הרצנו את התוכנית עבור קוד בגודל 5^5 וכן בלוח 5^7 , ראינו קפיצה משמעותית בזמני הריצה בכל פעם שהעלינו את גודל הלוח, כשניסינו להריץ על גודל 7^5 התוכנית נפלה בגלל זמן הריצה הגבוה

5. תוצאות:

תוצאות ההרצה עבור משחק בגודל 4^6 , ניתן לראות שזמן הריצה קטן, ממוצע של 0.6 כחצי שניה בנספח ניתן לראות תוצאות של 100 הרצות

[0 , 0 , 5 , 5]	0.5842154026031494	5
[5 , 5 , 2 , 3]	0.5842154026031494	4
[2 , 0 , 5 , 1]	0.13738226890563965	4
[2 , 0 , 1 , 2]	0.6203718185424805	4
[0 , 1 , 0 , 5]	0.3837730884552002	5
[5 , 4 , 4 , 5]	0.7575137615203857	5
[1 , 0 , 1 , 5]	0.8060715198516846	5
[5 , 5 , 5 , 0]	0.8410193920135498	5
[4 , 1 , 4 , 5]	0.8133575916290283	4

תוצאות ההרצה עבור משחק בגודל 5^5 ניתן לראות שזמן הריצה קטן, ממוצע של 3.05 שניות בנספח ניתן לראות תוצאות של 100 הרצות

[2 , 4 , 1 , 4 , 0]	3.9385523796081543	4
[0 , 1 , 1 , 3 , 2]	4.6977245807647705	5
[0 , 0 , 4 , 0 , 4]	2.3157904148101807	5
[4 , 0 , 4 , 3 , 3]	2.073746681213379	4
[4 , 2 , 1 , 4 , 4]	2.6813299655914307	5
[2 , 2 , 4 , 0 , 0]	2.7664122581481934	5
[0 , 4 , 1 , 2 , 2]	1.2841942310333252	5
[2 , 0 , 2 , 1 , 3]	2.0907015800476074	4
[1 , 3 , 2 , 3 , 3]	6.420062065124512	5
[1 , 3 , 0 , 0 , 2]	3.564898729324341	4

תוצאות עבור משחק בגודל 7^5 זמן ממוצע של 84.02 שניות כדקה וחצי

[0 , 3 , 3 , 3 , 5]	103.3285391330719	6
[5 , 3 , 4 , 3 , 5]	119.97155213356018	4
[0 , 4 , 0 , 0 , 2]	80.96241641044617	5
[6 , 0 , 5 , 3 , 1]	47.85110855102539	5
[4 , 3 , 6 , 6 , 3]	52.764320373535156	5
[2 , 2 , 4 , 0 , 0]	41.83356332778931	5
[5 , 2 , 3 , 4 , 4]	137.88456273078918	5
[0 , 0 , 1 , 1 , 6]	44.78705453872681	5
[2 , 0 , 2 , 0 , 4]	78.51684331893921	6
[0 , 4 , 3 , 6 , 4]	134.2802278995514	5

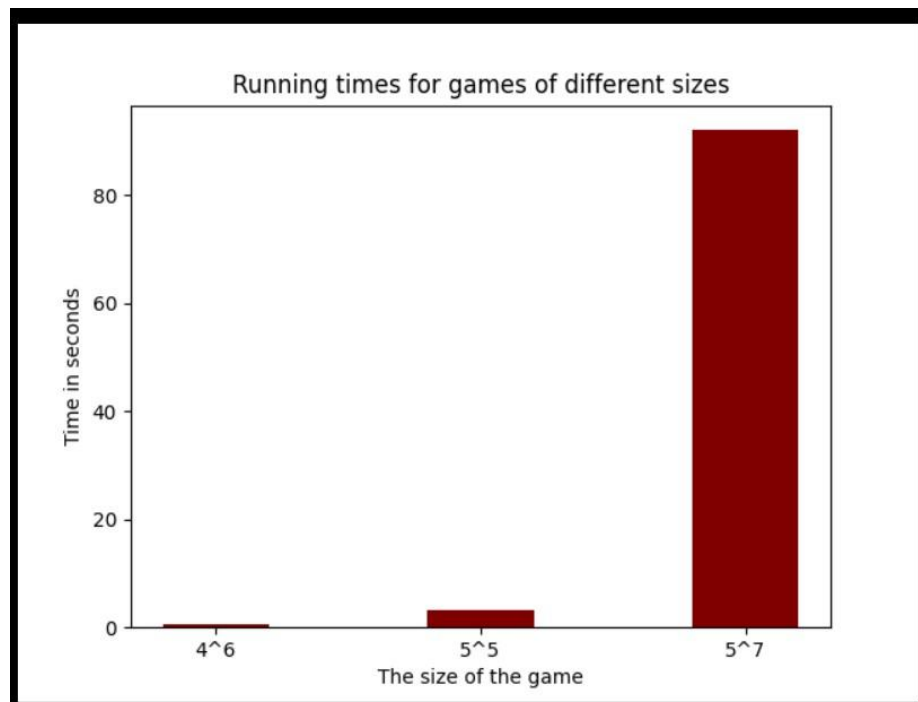
6. ניתוח התוצאות

ניתן לראות בטבלה שלהלן פתרון הבעיה המוצע עולה אקספוננציאלית. קיימת קפיצה גדולה בזמני הריצה בין הגדלים השונים של המשחק וכבר ב-7 מיקומים ו-10 צבעים לא יכולתי להגיע לפתרון בזמן סביר מספר הניחושים בין קוד באורך 4 לקוד באורך 5 לא עלה באופן משמעותי

להלן טבלת ממוצעי זמן ריצה ומספר ניחושים עבור 100 הרצות בכל משחק

מספר ניחושים ממוצע עד לפתרון	זמן ריצה ממוצע	גודל המשחק
4.55	0.64 כחצי שניה	4 מיקומים, 6 צבעים
4.42	3.22 שניות	5 מיקומים, 5 צבעים
5.26	99.02 כדקה וחצי	5 מיקומים, 7 צבעים
לא הגיע לפתרון זמן ארוך מידי	לא הגיע לפתרון זמן ארוך מידי	7 מיקומים, 10 צבעים

ניתן לראות את הפער הגדול בזמני הריצה...



7. ביקורת: בפרויקט רצינו לגלות את האסטרטגיה האופטימלית עבור בול

בפגיעה. ניתן לטעון שהאסטרטגיה האופטימלית היא ניחוש קוד בעל ההסתברות הגבוהה ביותר להתאמה. אולם עולה השאלה האם ישנם ניחושים שאינם אופטימליים מבחינה הסתברותית, איך מקדמים לכיוון הפתרון באופן מהיר יותר כמו שראינו בפתרון שהצגנו. משחק זה מאתגר ודורש שימוש בכלים של בינה מלאכותית משום שמרחב המצבים והפעולות גדול מאוד. מרחב המצבים הינו כל הניחושים הקודמים וכל הציונים שהם קיבלו. מספר האופציות לניחוש בודד הן n כפול m מספר הציונים האפשריים. מרחב המצבים האפשריים הוא דומה לקבוצת החזקה על מספר הניחושים 1 האפשריים. זאת קבוצת המצבים משום שהמשחקים הקודמים וציוניהם משפיעים על הניחושים הבאים. לכן נרצה למצוא פתרון שלא ירוץ על כלל האפשרויות אלא יבחר את האפשרות הטובה ביותר בזמן סביר כך שנוכל להריצו גם על קודים גדולים יחסית ולהגיע לפתרון קרוב

8. נספח:

פונקציה שמחזירה לי ניקוד עבור ניחוש כמה שחורים וכמה לבנים

```
def evaluate(guess, secret):
    code=list(secret) # we make a copy of c
    guess_copy=list(guess)
    well_placed, bad_placed =0,0
    for i,j in enumerate(guess_copy): #enum
        if j==code[i]: #well-placed
            well_placed+=1
            guess_copy[i]='#' #we do that i
            code[i]='*'
    for j in guess_copy:
        if j in code:
            bad_placed +=1
            code[code.index(j)]='*'
    evaluation=(well_placed, bad_placed)
    return evaluation
```

פונקציה שמחזירה לי קוד רנדומלי לזיהוי

```
[ ] def random_list(niv = 4, nb_color = 6): ## 4 and 6 are the values by default
    return [randint(0,nb_color-1) for i in range(niv)] # example secret=['1','4','4','5'] (the number are the colors)
```

פונקציה שמחזירה את כל הרצפים האפשריים עבור גודל קוד מסוים עם מספר הצבעים האפשרי

```
def possible_answers(niv=4,nb_color = 6):
    possible_answers= list(itertools.product(np.arange(0, nb_color, 1), repeat=niv))
    return np.array(possible_answers).tolist()
```

הפונקציה הירסטית

```
def heuristic(possible_answers, guess, eval_secret, S):

    #we create S
    new_list = S.copy()
    for ele in new_list:
        if evaluate(guess, ele) != eval_secret:
            S.remove(ele)

    #we create the table with score to all the possible answer

    max_score=[]

    for ele in possible_answers:
        dict_score={}
        for state in S:
            eval_score=evaluate(ele, state)
            if eval_score in dict_score:
                dict_score[(eval_score)]=dict_score[(eval_score)]+1

            else:
                dict_score[(eval_score)]=1

        max_value=0
        for score_key in dict_score:
            lenght=dict_score[(score_key)]
            if lenght>max_value:
                max_value=lenght

        max_score.append(max_value) #we have the maximum of each ele in possi

    best_guess=[]
    mini=1296 # all the possibilites
    for ele in max_score:
        if ele<mini:
            mini=ele
    for i in range(len(max_score)):
        if max_score[i]==mini:
            best_guess.append(possible_answers[i])
    for bg in best_guess:
        if bg in S:
            return bg, S
    for bg in best_guess:
        if bg in possible_answers:
            return bg, S
```

תוצאות ריצה עבור משחק בגודל 5 על 5

```
, [[0 , 1 , 2 , 3 , 0] , 5.0035881996154785 , 4]]
, [[1 , 3 , 4 , 2 , 0] , 2.325977087020874 , 4]
, [[4 , 3 , 1 , 0 , 0] , 1.4802021980285645 , 3]
, [[4 , 4 , 2 , 4 , 1] , 3.9913220405578613 , 4]
, [[0 , 1 , 3 , 0 , 3] , 6.188425302505493 , 5]
, [[4 , 1 , 1 , 0 , 1] , 2.267359972000122 , 4]
, [[0 , 0 , 3 , 3 , 3] , 4.799253225326538 , 5]
, [[0 , 1 , 0 , 4 , 2] , 3.4394609928131104 , 5]
, [[0 , 4 , 0 , 4 , 2] , 0.8484592437744141 , 3]
, [[0 , 4 , 0 , 1 , 3] , 4.859327793121338 , 5]
, [[4 , 4 , 2 , 4 , 4] , 0.06074357032775879 , 3]
, [[4 , 1 , 3 , 3 , 2] , 4.728121519088745 , 4]
, [[4 , 2 , 1 , 4 , 1] , 3.0226681232452393 , 4]
, [[2 , 3 , 3 , 0 , 3] , 3.1308395862579346 , 5]
, [[1 , 1 , 2 , 4 , 3] , 3.789909839630127 , 5]
, [[0 , 4 , 2 , 4 , 0] , 1.683908224105835 , 5]
, [[3 , 4 , 1 , 3 , 0] , 5.936312437057495 , 4]
, [[0 , 3 , 4 , 2 , 1] , 1.5322463512420654 , 4]
, [[4 , 4 , 4 , 1 , 3] , 3.228472948074341 , 5]
, [[4 , 0 , 0 , 4 , 2] , 5.670887231826782 , 4]
, [[0 , 0 , 1 , 1 , 2] , 2.1743979454040527 , 4]
, [[1 , 4 , 0 , 0 , 1] , 2.3886945247650146 , 5]
, [[1 , 2 , 3 , 1 , 4] , 1.4340286254882812 , 4]
, [[1 , 3 , 0 , 3 , 4] , 2.1261119842529297 , 5]
, [[0 , 3 , 4 , 2 , 3] , 5.493993043899536 , 5]
, [[0 , 2 , 3 , 0 , 1] , 3.197618007659912 , 4]
, [[4 , 0 , 1 , 4 , 0] , 3.8382155895233154 , 5]
, [[2 , 2 , 0 , 3 , 1] , 6.131333589553833 , 5]
, [[2 , 3 , 1 , 2 , 4] , 3.6919078826904297 , 4]
, [[4 , 3 , 2 , 4 , 1] , 1.5379843711853027 , 5]
, [[3 , 3 , 0 , 1 , 0] , 5.24162745475769 , 5]
, [[1 , 4 , 1 , 4 , 3] , 2.940885305404663 , 5]
, [[4 , 4 , 2 , 0 , 3] , 0.6452674865722656 , 4]
, [[1 , 2 , 0 , 1 , 0] , 3.7259247303009033 , 4]
, [[2 , 0 , 0 , 0 , 4] , 2.4645473957061768 , 4]
, [[0 , 0 , 2 , 3 , 4] , 0.7559866905212402 , 4]
, [[2 , 4 , 0 , 4 , 0] , 3.9837148189544678 , 5]
, [[1 , 4 , 4 , 0 , 1] , 3.0116186141967773 , 5]
, [[1 , 4 , 0 , 4 , 1] , 2.4772472381591797 , 4]
, [[3 , 4 , 0 , 4 , 4] , 1.6919569969177246 , 5]
, [[3 , 4 , 2 , 0 , 4] , 2.9392898082733154 , 4]
, [[3 , 1 , 3 , 2 , 2] , 4.919798851013184 , 5]
, [[4 , 3 , 1 , 4 , 1] , 2.2264528274536133 , 3]
, [[1 , 0 , 1 , 2 , 1] , 3.6518521308898926 , 4]
, [[3 , 0 , 1 , 0 , 2] , 7.296464443206787 , 5]
, [[2 , 0 , 4 , 3 , 0] , 4.835401773452759 , 4]
, [[0 , 3 , 1 , 2 , 4] , 0.2558448314666748 , 5]
, [[0 , 3 , 3 , 1 , 1] , 3.704155445098877 , 5]
```

, [[4 ,2 ,1 ,4 ,1] ,4.9918739795684814 ,5]
, [[0 ,4 ,2 ,3 ,0] ,2.231372356414795 ,4]
, [[3 ,2 ,2 ,1 ,3] ,0.9149315357208252 ,4]
, [[2 ,1 ,0 ,4 ,3] ,5.633694887161255 ,6]
, [[2 ,2 ,3 ,2 ,1] ,2.544029474258423 ,5]
, [[2 ,4 ,1 ,0 ,3] ,2.1703522205352783 ,4]
, [[4 ,1 ,0 ,2 ,0] ,0.9453165531158447 ,4]
, [[4 ,1 ,1 ,4 ,1] ,1.934386968612671 ,5]
, [[3 ,4 ,1 ,0 ,3] ,6.034074306488037 ,5]
, [[3 ,3 ,1 ,1 ,0] ,4.240875482559204 ,5]
, [[1 ,1 ,0 ,2 ,0] ,2.616206407546997 ,4]
, [[0 ,4 ,1 ,0 ,0] ,2.776076555252075 ,3]
, [[4 ,2 ,4 ,0 ,0] ,5.068762540817261 ,4]
, [[4 ,3 ,4 ,4 ,0] ,1.0767242908477783 ,5]
, [[1 ,1 ,3 ,1 ,2] ,2.4519073963165283 ,4]
, [[3 ,1 ,2 ,0 ,4] ,3.6779019832611084 ,5]
, [[0 ,3 ,1 ,3 ,4] ,5.154526948928833 ,4]
, [[3 ,2 ,3 ,0 ,1] ,2.49664044380188 ,5]
, [[3 ,3 ,3 ,2 ,2] ,1.0718860626220703 ,4]
, [[0 ,1 ,1 ,1 ,2] ,0.8509106636047363 ,4]
, [[2 ,3 ,3 ,0 ,1] ,4.738630533218384 ,4]
, [[1 ,0 ,2 ,4 ,1] ,5.034293174743652 ,5]
, [[0 ,2 ,4 ,2 ,0] ,2.6498706340789795 ,5]
, [[3 ,2 ,2 ,0 ,2] ,2.5042247772216797 ,4]
, [[2 ,2 ,1 ,2 ,3] ,3.7817015647888184 ,5]
, [[3 ,1 ,2 ,0 ,4] ,1.5473883152008057 ,5]
, [[3 ,4 ,0 ,0 ,0] ,0.45348310470581055 ,3]
, [[0 ,1 ,3 ,0 ,4] ,2.246978759765625 ,5]
, [[2 ,3 ,2 ,2 ,0] ,3.536557912826538 ,4]
, [[3 ,2 ,1 ,1 ,3] ,2.854642868041992 ,5]
, [[3 ,1 ,3 ,3 ,3] ,5.6321728229522705 ,5]
, [[3 ,2 ,2 ,2 ,2] ,1.5948801040649414 ,4]
, [[1 ,4 ,0 ,4 ,1] ,2.3007454872131348 ,4]
, [[3 ,2 ,4 ,0 ,4] ,5.096553325653076 ,5]
, [[3 ,2 ,2 ,2 ,0] ,2.392230272293091 ,4]
, [[3 ,4 ,1 ,2 ,4] ,2.3601269721984863 ,5]
, [[1 ,1 ,4 ,1 ,4] ,1.8668630123138428 ,5]
, [[1 ,0 ,3 ,1 ,3] ,2.4934403896331787 ,5]
, [[1 ,1 ,3 ,2 ,3] ,4.546716690063477 ,5]
, [[1 ,2 ,4 ,0 ,0] ,2.5151572227478027 ,5]
, [[3 ,3 ,0 ,3 ,3] ,4.80941104888916 ,5]
, [[2 ,1 ,0 ,3 ,4] ,4.750749111175537 ,5]
, [[4 ,4 ,2 ,1 ,2] ,4.222819089889526 ,4]
, [[3 ,0 ,0 ,4 ,4] ,4.683806657791138 ,5]
, [[4 ,4 ,1 ,1 ,0] ,5.28309178352356 ,5]
, [[3 ,0 ,0 ,3 ,2] ,2.3583550453186035 ,5]
, [[4 ,4 ,2 ,4 ,0] ,0.8990964889526367 ,5]
, [[0 ,3 ,2 ,0 ,1] ,4.89142918586731 ,5]
, [[3 ,3 ,4 ,2 ,0] ,2.386388063430786 ,5]
, [[3 ,3 ,2 ,3 ,0] ,2.834946870803833 ,4]
, [[2 ,0 ,2 ,1 ,4] ,4.743038654327393 ,5]

[[[0 ,0 ,2 ,1 ,4] ,5.1242311000823975 ,5]

תוצאות ריצה עבור ניחוש בגודל 4 עם 6 צבעים

, [[5 ,2 ,5 ,3] ,0.4612274169921875 ,5]]
, [[3 ,5 ,0 ,4] ,0.776104211807251 ,5]
, [[5 ,4 ,1 ,5] ,0.32199907302856445 ,4]
, [[1 ,5 ,1 ,4] ,0.9083347320556641 ,5]
, [[5 ,4 ,4 ,4] ,0.711174726486206 ,4]
, [[4 ,5 ,1 ,1] ,0.6607768535614014 ,4]
, [[3 ,0 ,3 ,3] ,0.25045180320739746 ,4]
, [[5 ,5 ,2 ,1] ,0.7601127624511719 ,4]
, [[3 ,3 ,1 ,4] ,0.8284096717834473 ,5]
, [[0 ,4 ,5 ,3] ,0.4187180995941162 ,3]
, [[0 ,2 ,2 ,5] ,1.6819519996643066 ,5]
, [[2 ,3 ,3 ,0] ,1.3610420227050781 ,5]
, [[3 ,2 ,3 ,2] ,0.1334688663482666 ,4]
, [[3 ,5 ,4 ,3] ,0.7721588611602783 ,5]
, [[0 ,0 ,3 ,1] ,0.2947702407836914 ,4]
, [[0 ,0 ,5 ,5] ,0.650005578994751 ,4]
, [[3 ,5 ,3 ,4] ,0.5426888465881348 ,5]
, [[5 ,5 ,4 ,1] ,0.31902527809143066 ,4]
, [[0 ,3 ,2 ,4] ,0.07924842834472656 ,4]
, [[2 ,0 ,0 ,5] ,0.5374560356140137 ,5]
, [[3 ,4 ,0 ,2] ,0.8254203796386719 ,5]
, [[1 ,5 ,2 ,5] ,0.3333446979522705 ,4]
, [[4 ,0 ,2 ,3] ,0.4137411117553711 ,5]
, [[2 ,5 ,3 ,4] ,0.33376431465148926 ,5]
, [[5 ,1 ,3 ,1] ,0.5232019424438477 ,5]
, [[5 ,5 ,4 ,3] ,0.4640803337097168 ,4]
, [[3 ,2 ,2 ,4] ,0.17493295669555664 ,4]
, [[5 ,0 ,1 ,2] ,0.5956189632415771 ,4]
, [[0 ,5 ,0 ,4] ,0.732081413269043 ,5]
, [[0 ,2 ,0 ,0] ,0.8302674293518066 ,5]
, [[2 ,4 ,2 ,4] ,0.5372586250305176 ,4]
, [[5 ,5 ,5 ,1] ,0.3162655830383301 ,4]
, [[2 ,0 ,4 ,2] ,0.30667614936828613 ,4]
, [[2 ,0 ,2 ,1] ,1.4114363193511963 ,3]
, [[2 ,4 ,5 ,0] ,1.3041105270385742 ,5]
, [[2 ,1 ,3 ,0] ,0.667320728302002 ,4]
, [[1 ,1 ,1 ,0] ,0.25643062591552734 ,4]
, [[3 ,1 ,4 ,1] ,0.6964960098266602 ,4]
, [[2 ,5 ,5 ,5] ,0.3067891597747803 ,5]
, [[2 ,3 ,5 ,5] ,0.2588934898376465 ,5]
, [[1 ,0 ,5 ,4] ,0.14008665084838867 ,4]
, [[5 ,3 ,4 ,2] ,0.2457573413848877 ,4]
, [[4 ,5 ,2 ,3] ,0.6779444217681885 ,5]
, [[0 ,2 ,0 ,2] ,0.9254646301269531 ,4]

, [[2 ,1 ,5 ,2] ,0.6432969570159912 ,4]
, [[3 ,3 ,4 ,0] ,0.7702813148498535 ,4]
, [[5 ,2 ,1 ,3] ,0.7432975769042969 ,4]
, [[5 ,4 ,3 ,0] ,0.3139941692352295 ,4]
, [[3 ,1 ,1 ,4] ,0.4196786880493164 ,4]
, [[3 ,3 ,4 ,1] ,0.25889134407043457 ,4]
, [[5 ,4 ,5 ,1] ,0.899397611618042 ,5]
, [[0 ,3 ,2 ,1] ,0.9199950695037842 ,5]
, [[1 ,3 ,4 ,4] ,0.9266166687011719 ,5]
, [[4 ,2 ,3 ,2] ,0.291154146194458 ,4]
, [[2 ,2 ,0 ,3] ,1.5561909675598145 ,5]
, [[0 ,2 ,1 ,0] ,1.2650997638702393 ,4]
, [[0 ,2 ,0 ,2] ,0.4704091548919678 ,4]
, [[2 ,3 ,0 ,5] ,0.3293135166168213 ,4]
, [[1 ,1 ,3 ,0] ,0.2649381160736084 ,4]
, [[2 ,2 ,0 ,0] ,0.723703145980835 ,5]
, [[5 ,3 ,3 ,5] ,0.7749481201171875 ,5]
, [[4 ,0 ,3 ,4] ,0.7808923721313477 ,3]
, [[0 ,1 ,0 ,5] ,0.22217535972595215 ,3]
, [[3 ,4 ,5 ,2] ,0.6524672508239746 ,5]
, [[5 ,3 ,0 ,4] ,0.8324558734893799 ,5]
, [[3 ,3 ,1 ,0] ,0.8001952171325684 ,4]
, [[5 ,3 ,1 ,3] ,0.3052687644958496 ,4]
, [[3 ,3 ,1 ,0] ,0.7682130336761475 ,4]
, [[3 ,3 ,0 ,5] ,0.924095630645752 ,5]
, [[0 ,1 ,1 ,2] ,0.9078652858734131 ,5]
, [[4 ,0 ,5 ,3] ,0.6894302368164062 ,5]
, [[5 ,4 ,2 ,3] ,0.7765233516693115 ,5]
, [[4 ,5 ,5 ,3] ,1.1034507751464844 ,5]
, [[3 ,3 ,1 ,2] ,1.2361078262329102 ,4]
, [[1 ,4 ,5 ,5] ,0.319293737411499 ,5]
, [[2 ,5 ,0 ,2] ,0.4609837532043457 ,5]
, [[4 ,2 ,3 ,5] ,0.45966315269470215 ,5]
, [[4 ,0 ,1 ,4] ,0.8031456470489502 ,5]
, [[2 ,5 ,2 ,0] ,0.7102148532867432 ,5]
, [[2 ,3 ,3 ,0] ,0.8642160892486572 ,5]
, [[3 ,5 ,0 ,5] ,0.3118588924407959 ,4]
, [[2 ,4 ,2 ,3] ,0.7551758289337158 ,5]
, [[2 ,2 ,0 ,0] ,0.7146711349487305 ,5]
, [[2 ,3 ,0 ,4] ,0.25349950790405273 ,5]
, [[1 ,2 ,3 ,2] ,0.7874605655670166 ,5]
, [[4 ,5 ,3 ,2] ,0.9364902973175049 ,5]
, [[3 ,4 ,4 ,4] ,0.6826090812683105 ,5]
, [[2 ,3 ,2 ,1] ,0.526982307434082 ,4]
, [[4 ,3 ,2 ,5] ,0.7588012218475342 ,5]
, [[4 ,1 ,2 ,0] ,0.7795884609222412 ,4]
, [[2 ,1 ,3 ,2] ,0.2639648914337158 ,4]
, [[2 ,3 ,3 ,0] ,0.9605159759521484 ,3]
, [[4 ,0 ,0 ,2] ,1.2890732288360596 ,4]
, [[2 ,5 ,4 ,3] ,0.3445417881011963 ,5]
, [[4 ,5 ,0 ,3] ,0.7329318523406982 ,4]

```
, [[3 ,1 ,2 ,1] ,0.5682551860809326 ,4]
, [[1 ,1 ,1 ,4] ,0.6289002895355225 ,4]
, [[4 ,3 ,2 ,5] ,0.6605772972106934 ,5]
, [[4 ,3 ,2 ,2] ,0.7587113380432129 ,5]
[[[1 ,0 ,5 ,0] ,0.5892157554626465 ,4]
```

עבור משחק בגודל 7 5

```
, [[6 ,5 ,0 ,6 ,0] ,124.91638350486755 ,5]]
, [[0 ,1 ,3 ,1 ,0] ,123.85765767097473 ,4]
, [[2 ,6 ,1 ,1 ,0] ,135.37129855155945 ,6]
, [[5 ,0 ,2 ,2 ,1] ,132.4817190170288 ,5]
, [[0 ,1 ,6 ,2 ,1] ,136.190327167511 ,6]
, [[0 ,3 ,0 ,2 ,2] ,35.09770464897156 ,5]
, [[6 ,3 ,5 ,6 ,1] ,79.52936148643494 ,5]
, [[2 ,2 ,0 ,1 ,5] ,159.26262617111206 ,6]
, [[2 ,0 ,1 ,4 ,0] ,148.36331176757812 ,5]
, [[3 ,6 ,2 ,2 ,0] ,94.11705756187439 ,4]
, [[4 ,3 ,3 ,0 ,0] ,162.45776963233948 ,5]
, [[5 ,6 ,1 ,1 ,1] ,10.290385484695435 ,4]
, [[5 ,3 ,5 ,5 ,3] ,40.06660056114197 ,5]
, [[6 ,3 ,6 ,1 ,0] ,85.36517715454102 ,4]
, [[1 ,2 ,0 ,0 ,6] ,18.785879135131836 ,5]
, [[2 ,4 ,5 ,0 ,0] ,153.6961534023285 ,5]
, [[1 ,5 ,6 ,1 ,6] ,46.959476470947266 ,5]
, [[4 ,0 ,4 ,3 ,1] ,132.33136796951294 ,5]
, [[5 ,1 ,5 ,0 ,2] ,43.6345009803772 ,5]
, [[4 ,5 ,2 ,4 ,1] ,51.17653250694275 ,5]
, [[4 ,6 ,1 ,1 ,3] ,148.47332453727722 ,6]
, [[0 ,0 ,5 ,0 ,3] ,147.8437523841858 ,6]
, [[6 ,1 ,0 ,5 ,0] ,79.63026165962219 ,5]
, [[6 ,3 ,0 ,0 ,3] ,79.02158212661743 ,4]
, [[3 ,5 ,6 ,0 ,0] ,149.5408148765564 ,6]
, [[1 ,4 ,6 ,1 ,6] ,38.15278100967407 ,5]
, [[6 ,3 ,4 ,4 ,2] ,46.535808086395264 ,5]
, [[3 ,4 ,3 ,0 ,3] ,136.46400022506714 ,6]
, [[4 ,4 ,1 ,1 ,3] ,76.2466561794281 ,4]
[[[3 ,1 ,6 ,0 ,5] ,17.153923988342285 ,5]
, [[4 ,0 ,6 ,0 ,4] ,96.52748322486877 ,6]]
, [[1 ,2 ,2 ,6 ,3] ,138.3396120071411 ,6]
, [[3 ,2 ,5 ,6 ,4] ,128.86575651168823 ,6]
, [[0 ,2 ,2 ,3 ,3] ,48.87163782119751 ,6]
, [[5 ,2 ,3 ,1 ,5] ,147.08650922775269 ,5]
, [[0 ,0 ,3 ,2 ,4] ,115.74406051635742 ,5]
, [[0 ,6 ,0 ,6 ,5] ,125.50348663330078 ,6]
, [[0 ,0 ,4 ,0 ,6] ,107.44255208969116 ,6]
, [[6 ,5 ,3 ,5 ,4] ,118.2224349975586 ,5]
, [[0 ,2 ,0 ,1 ,6] ,164.80814242362976 ,6]
, [[1 ,4 ,6 ,4 ,4] ,38.372743368148804 ,5]
, [[0 ,1 ,5 ,0 ,3] ,159.07315826416016 ,6]
```

, [[6, 5, 5, 0, 4], 165.75499439239502, 6]
, [[1, 5, 2, 4, 3], 147.49840664863586, 6]
, [[1, 4, 4, 0, 2], 48.13024306297302, 5]
, [[4, 5, 0, 6, 1], 2.117105484008789, 5]
, [[5, 0, 3, 6, 2], 153.64235877990723, 6]
, [[5, 1, 3, 1, 6], 146.68642377853394, 6]
, [[6, 1, 0, 1, 5], 46.59438610076904, 6]
, [[0, 1, 3, 0, 3], 131.2932732105255, 5]
, [[2, 3, 6, 3, 3], 11.986742973327637, 5]
, [[4, 4, 3, 3, 4], 45.49861526489258, 5]
, [[0, 0, 6, 3, 3], 38.915018796920776, 5]
, [[6, 1, 1, 1, 3], 44.559977769851685, 5]
, [[5, 4, 4, 6, 6], 130.93221306800842, 6]
, [[4, 2, 2, 1, 0], 126.89867401123047, 6]
, [[2, 4, 1, 5, 3], 42.06775951385498, 6]
, [[3, 5, 4, 4, 1], 140.4376084804535, 4]
, [[4, 3, 1, 5, 3], 163.1763777732849, 6]
[[[1, 6, 0, 0, 4], 148.77630639076233, 5]
, [[4, 6, 4, 1, 1], 167.22391033172607, 6]]]
, [[2, 0, 4, 0, 6], 121.52723288536072, 6]
, [[6, 0, 6, 2, 6], 78.14880418777466, 5]
, [[5, 4, 4, 1, 5], 85.89476203918457, 6]
, [[1, 4, 1, 3, 0], 67.12408232688904, 5]
, [[4, 5, 5, 2, 2], 12.658886671066284, 5]
, [[2, 4, 3, 3, 6], 86.77457046508789, 6]
, [[6, 3, 5, 0, 0], 37.050267934799194, 5]
, [[4, 3, 4, 6, 1], 76.1837420463562, 5]
, [[5, 3, 2, 6, 2], 109.60022926330566, 6]
, [[3, 1, 2, 3, 5], 139.2237594127655, 6]
, [[4, 1, 1, 3, 2], 154.36710810661316, 5]
, [[4, 2, 5, 0, 5], 148.49872136116028, 6]
, [[2, 3, 4, 4, 1], 113.82541298866272, 5]
, [[5, 6, 3, 3, 1], 167.71689176559448, 4]
, [[6, 6, 0, 0, 5], 86.40008592605591, 6]
, [[0, 5, 3, 4, 1], 130.3680899143219, 5]
, [[0, 1, 1, 2, 1], 12.380309104919434, 4]
, [[3, 4, 0, 3, 3], 130.06645154953003, 6]
, [[0, 1, 4, 4, 1], 92.97621321678162, 4]
, [[3, 0, 5, 1, 1], 109.50776433944702, 5]
, [[2, 6, 2, 1, 2], 17.581724882125854, 5]
, [[0, 2, 4, 1, 0], 142.9277627468109, 4]
, [[4, 3, 0, 4, 3], 35.51115417480469, 5]
, [[4, 1, 2, 3, 5], 154.27764081954956, 6]
, [[2, 1, 3, 0, 3], 86.33786630630493, 6]
, [[1, 2, 5, 0, 6], 21.99136519432068, 5]
, [[3, 1, 4, 3, 0], 161.4455063343048, 6]
, [[5, 6, 5, 4, 0], 117.19215154647827, 5]
[[[2, 6, 0, 5, 4], 90.54495239257812, 5]

9. מקורות:

[בול פגיעה – ויקיפדיה](#)

[the computer as master mind - donald e. knuth](#)

[Moodle](#)