



הפקולטה להנדסה  
ע"ש אלכסנדר קופקין  
אוניברסיטת ברי"א



בואו לשנות  
את המחר.  
היום.

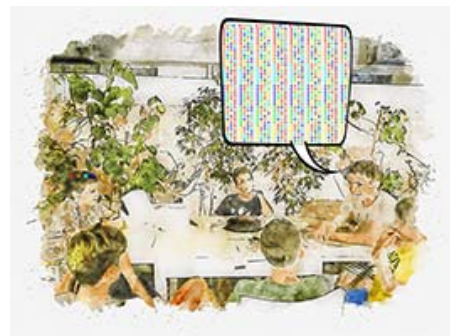
פתרון חידה

מס' 2

בניזולטר הקודם שאלנו מהו המספר המינימלי של צעדים שידרשו על מנת לנחש נכונה רצף של 5 ספרות מתוך 9 אפשרויות. התשובה הקצרה לשאלה זו היא 9. ניתן לחשב את התשובה באמצעות הקוד

בקישור-[https://colab.research.google.com/drive/1x7SyXEJUx-p3qXh\\_N3dEhgB\\_1JJRLmU9?usp=sharing](https://colab.research.google.com/drive/1x7SyXEJUx-p3qXh_N3dEhgB_1JJRLmU9?usp=sharing)

ננסה להסביר את התשובה בקצרה.



חידה זו היא דוגמה קלסית למקרה שהפתרון מצריך יכולת אלגוריתמית וקידוד. אני לא מכיר דרך לפתור את החידה לא שימוש במחשב.

באופן כללי מוספר הקודים האפשריים הוא

$$s_{n,k} = \binom{n}{k} k!$$

נסמן ב  $s_{n,k}$  את קבוצת כל הקודים החוקים.

כאשר  $n$  זה מספר האותיות  $k$ - זה אורך הקוד אצלנו בכתב החידה

$$n = 9, k = 5$$

כלומר

$$s_{9,5} = 15120$$

כעת אפשר לשאול את עצמנו באיזה סדר אנחנו הולכים לנחש את הקוד. אפשר לחשוב שאפשר לסדר את השאלות לפני שחישבנו את התשובה, אבל במקרה זה מספר האפשרויות לסדר את השאלות שלנו הוא מספר אסטרונומי!  $s_{n,k}$

נסמן קבוצה זאת קבוצת כל הפרמוטציות של הקודים החוקים ב-  $\rho_{n,k}$ .

כעת נדמיין כי השחקן שמנסה לנחש את הקוד מגלה לנו את האסטרטגיה שלו. הוא אומר כי הוא הולך לשאול את השאלות בסדר של הפרמוטציות  $\pi$  כל עוד הם לא סותרות את התשובות שקבלנו קודם. באופן כללי השאלה הראשונה תהיה  $\pi_1$

הבעיה היא שאנחנו עדיין לא יכולים לפתור את השאלה כי הקבוצה  $\rho_{n,k}$  גדולה מידי. ולכן זמן החישוב יהיה גדול מידי. כדי להתגבר על מגבלה זו נשים לב כי ניתן לבחור כל פרמוטציה  $\pi \in \rho_{n,k}$  והתשובה לא תשתנה (זה נובע משקולי סימטריה)

נבדוק מה הדוגמה הבאה מלמדת אותנו.

אם נרצה לכתוב תוכנית מחשב שמחשבת את מספר השאלות אנו צריכים:

1. פונקציה שמחשבת את כל הקודים האפשריים בקישור שלנו הפונקציות הם:

- a. `import itertools`
- b. `z=list(itertools.permutations([1,2,3,4,5,6,7,8,9],5))`
- c. `CalculationNumberHits`
- d. `CalculateExactNumber`

2. פונקציה שמחשבת את הקודים האפשריים. בקישור שלנו

- a. `ReducingTheNumberOfPossibleCodes`

3. פונקציה שמקבלת סדר של שאלות וקוד ומחשבת את מספר השאלות לניחוש הקוד בסדר שאלות נתון. בקישור שלנו

- a. `FindCode`

4. מציאת המקרה שדורש הכי הרבה שאלות. בקישור שלנו

5. זה נמצא בתא האחרון.

באופן כללי אפשר להגדיר פונקציה המחשבת את מספר השאלות שאלגוריתם שאל עבור הקוד `code` והפרמוטציה  $\pi$ . נסמן את הפונקציה הזאת  $numQ(code, \pi)$ . באמצעות פונקציה זאת אנו יכולים רשום את מה שאנו רוצים לחשב

$$\max_{code \in S_{n,k}, \pi \in \rho_{n,k}} numQ(code, \pi)$$

אנחנו, כאמור, בחרנו בפרמוטציה  $\pi \in \rho_{n,k}$  ולכן אנו רוצים לחשב את

$$\max_{code \in S_{n,k}} numQ(code, \pi)$$

דוגמה לקוד שמצריך 9 שאלות ובסדר הסטנדרטי של python ניתן לראות

```

the code is (6, 4, 7, 3, 8)
question Code is (1, 2, 3, 4, 5)
h1= 2
n1= 0
len= 3120
question Code is (2, 1, 6, 7, 8)
h1= 3
n1= 1
len= 582
question Code is (2, 3, 7, 6, 9)
h1= 3
n1= 1
len= 91
question Code is (2, 4, 8, 9, 6)
h1= 3
n1= 1
len= 13
question Code is (2, 5, 9, 8, 7)
h1= 2
n1= 0
len= 10
question Code is (6, 1, 7, 9, 4)
h1= 3

```

```

n1= 2
len= 4
question Code is (6, 1, 8, 3, 9)
h1= 3
n1= 2
len= 2
question Code is (6, 3, 8, 7, 4)
h1= 5
n1= 1
len= 1
question Code is (6, 4, 7, 3, 8)
h1= 5
n1= 5
len= 1
the code is (6, 4, 7, 3, 8)
[(6, 4, 7, 3, 8), 9]

```

## מצורף הקוד

```

import itertools
z=list(itertools.permutations([1,2,3,4,5,6,7,8,9],5))
def CalculationNumberHits(code1,code2):
    return len(set(code1).intersection(set(code2)))

def CalculateExactNumber(code1,code2):
    return sum(i1 == i2 for i1, i2 in zip(code1, code2))

def ReducingTheNumberofPossibleCodes(listofcode,code1,numHits,numExacr)
:
    newList=[]
    for i in listofcode:
        if CalculationNumberHits(i,code1)==numHits and CalculateExactNu
mber(i,code1)==numExacr:
            newList.append(i)
    return newList

def FindCode(allLegalPermutations,code1):
    bol=True
    ans=0
    #print("the code is",code1)
    legalPermutations=allLegalPermutations
    while bol:
        ans=ans+1
        questionCode=legalPermutations[0]
        if questionCode==code1:
            bol=False
        # print("question Code is",questionCode)
        h1=CalculationNumberHits(code1,questionCode)
        # print("h1=", h1)
        n1=CalculateExactNumber(code1,questionCode)
        # print("n1= ", n1)

```

```
        legalPermutations=ReducingTheNumberofPossibleCodes(legalPermuta
tions,questionCode,h1,n1)
        # print("len=",len(legalPermutations))
        #print("the code is ",questionCode)
        return [questionCode,ans]
```

```
ans=[FindCode(z,i) for i in z ]
n=[i[1] for i in ans]
print(n[n.index(max(n))])
```