

קומבינציות של סידרת מספרים

1. מתחילים עם מערך של 2 מערכים a (כל האפשרויות לסידור 2 מספרים: $[[1,2],[2,1]]$)
2. לולאה מ-3 ועד למספר שרוצים לבנות מערך חדש, כל פעם ב-1 יותר. האורך של החדש ב-1 יותר. מכינים את המשתנים עבור אורך ישן, חדש ומערך של מערכים `new_a`.
3. לולאה כגודל המערך הישן (למשל 2 בהתחלה). כעת נגדיר מערך כגודל המספר החדש b (אחד מעל הקיים) שיכיל בהתחלה אפסים.
4. לולאה כגודל המערך החדש, שבה נציב בכל פעם את המספר החדש בתא הבא של b ונערבב אותו עם המערך הקודם. לדוגמה:

$[[1,2],[2,1]] + [0,0,0]$

העירבוב נעשה כאשר שמים את המספר החדש בכל פעם במיקום אחר עם המערכים הקיימים (לפני כל התאים, ביניהם ואחריהם), למשל:

$3,[1,2]$, $3,[2,1]$,
 $[1,3,2]$, $[2,3,1]$,
 $[1,2],3$, $[2,1],3$

הטכניקה היא על ידי קביעת b כלמשל במקרה של 3:

$[3,0,0]$, $[0,3,0]$, $[0,0,3]$

ובכל סיבוב של הלולאה, יחליפו המספרים מהמערכים: $[1,2]$ ו- $[2,1]$ את האפסים כמו בדוגמה.

מספר המערכים תמיד יגדל פי המספר התורן, כלומר מ-2 יהיה 6 (2×3), מ-6 יהיה 24 (6×4), מ-24 יהיה 120 (24×5) וזה אכן תואם למספר הצירופים שהוא עצרת מספר האפשרויות.

הנה דוגמה לקוד אפשרי בפייתון:

```

def main():
    n=int(input('enter number of permutations (3 or more): '))
    permutations=[[1,2],[2,1]]          # The base combo of n=2
    for length in range(3,n+1):        # from 3 up
        old_len=len(permutations[0])   # Length of existing chain
        new_len=old_len+1              # Length of new chain
        " permutations will hold the entire new set of combos and b holds"
        " each sequence in turn (out of the entire set) "
        new_permutations = []
        for s in range(len(permutations)):
            b=[0]*length
            for j in range(new_len):
                b[j]=length
            for k in range(old_len):
                for k1 in range(new_len):
                    if b[k1]==0:
                        b[k1]=permutations[s][k]
                        break
            new_permutations = new_permutations + [b]
            b=[0]*length #init b to zeroes so only the one in turn
                        #is the new number added

        count_p=0
        if length == n:
            for row in new_permutations:
                print(row)
                count_p +=1
            print('above combinations are: ', count_p)
            permutations=new_permutations
        print(permutations)
if __name__ == '__main__':
    main()

```