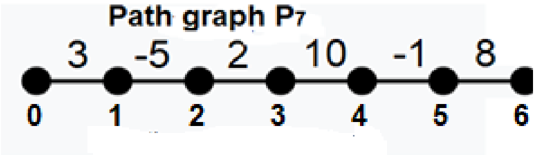
שעור 3 מציאת קטע (רצוף) בתוך מערך בעל סכום איברים גדול ביותר

בהינתן מערך של מספרים יש למצוא קטע (רצוף) בעל סכום איברים גדול ביותר. את המערך ניתן להציג כגרף המסלול (path graph or linear graph).

גרף המסלול הוא גרף בעל n קדקודים שהוא מסלול פשוט ומסומן על ידי P_n.

לדוגמה, מערך `a[] = {3, -5, 2, 10, -1, 8}` ניתן להציג כגרף מסלול עם משקלים על הצלעות:



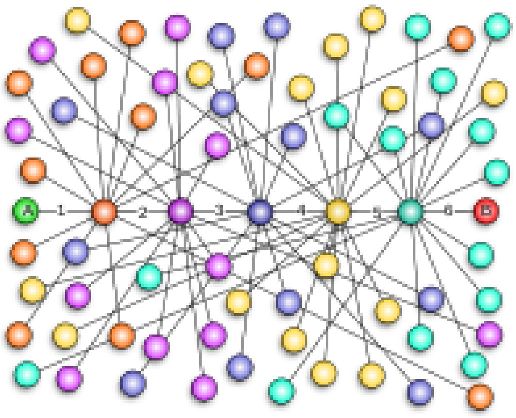
נגדיר את קוטר הגרף:

אורך המסלול בגרף – הוא שווה למספר הצלעות לאורכו.

המרחק בין שני קדקודי הגרף מוגדר כאורך המזערי של מסלול ביניהם.

קוטר הגרף – הוא המרחק המקסימאלי בגרף בין זוג קדקודים כלשהם.

קוטר הגרף מתקשר ל**תופעת העולם הקטן** (small world problem), לפיה כל אדם יכול ליצור קשר עם כל אדם אחר בעולם דרך מספר קטן של מתווכים.



"שש דרגות של הפרדה "הוא מושג בתרבות הפופולרית שנובע וקשור באופן הדוק לתופעת העולם הקטן. כמוצג באיור, משמעותו היא שניתן ליצור קשר בין כל שני אנשים באמצעות שש היכרויות (או לחלופין, חמישה אנשי קשר) בלבד.

דוגמה: אליזבט - דיקן הפקולטה - רקטור האוניברסיטה – נגיד האוניברסיטה – ראש הממשלה – נשיא ארה"ב.

ניתן לומר שמרחק בין שני אנשים ברשת הסוציאלית בממוצע הוא 6. במילים אחרות ניתן לומר שקוטר הגרף שמייצג את הרשת הסוציאלית הוא מספר שקרוב ל-6.

מציאת קטע (רצוף) בעל סכום איברים גדול ביותר במערך נתון

חיפוש שלם. מספר קטעים רצופים במערך הוא :

$$(n-1)+(n-2)+\ldots+1=\frac{n(n-1)}{2}=O(n^2)$$

סיבוכיות חישוב סכום של קטע היא $O(n)$, לכן סיבוכיות של חיפוש שלם היא $\mathbf{O(n)*O(n^2)=O(n^3)}$.

אלגוריתם חמדני – לא עובד.

תכנות דינאמי. נחשב את סכומים של כל $\frac{n(n-1)}{2}$ ונשמור אתם במטריצה בגודל $n*n$.

את סכום של הקטע $[i,j]$ נחשב על סמך סכום של קטע $[i,j-1]$.

נגדיר מטריצה $S[n][n]$, באופן הבא:

$S[i][i] = a[i],\ \ S[i][j] = S[i][j-1] + S[j][j] = S[i][j-1] + a[j]$

דוגמה: $\text{a}[] \ = \ \{3, -2, 5, 1\}$

	1	2	3	4
1	3	1	6	7
2		-2	3	4
3			5	6
4				1

הסכום המקסימאלי הוא 7, הקטע בעל סכום מקסימאלי הוא [1,4], כלומר סכום של כל איברי המערך:

$7 = 3 - 2 + 5 + 1$

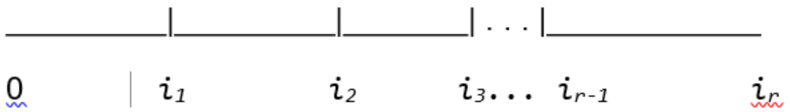
סיבוכיות האלגוריתם היא $O(n^2)$.

אלגוריתם אופטימאלי שנקרא best ועובד ב-O(n).

תכנות דינאמי נותן $\frac{n(n-1)}{2} \approx \frac{n^2}{2}$ קטעים וביניהם אנו בוחרים בקטע בעל סכום מקסימאלי. עכשיו נפרק את המערך ל-**n** ונוכיח שלא מדלגים על קטעים-מועמדים למקסימליים.

1. מתחילים לסכום את איברי המערך מאיבר חיובי ראשון.
2. סכמים את האיברים ושומרים על הסכום המקסימאלי עד שנקבל סכום שלילי. כלומר בכל תת-קטע $[a_i, \dots, a_k]$ הסכומים $[a_i + a_{i+1} + \dots, a_j > 0]$ לכל $j < k$.
3. מאפסים את הסכום וחוזרים לשלב 2.

קבלנו חלוקת המערך לתתי-קטעים ומספר הקטעים קטן או שווה n.



טענה1. קטע בעל סכום מקסימאלי הוא מהקטעים האלה.

הוכחה. שימו לב כי המספר האחרון של כל תת-מרווח הוא שלילי, בגלל שלפניו הסכום היה חיובי. המספר הראשון של כל תת-משנה הוא חיובי, אחרת הוא בעצמו סכום שלילי, ואנחנו מדלגים עליו. אם מתחיל ים לסכום מאיבר חיובי (אבל לא הראשון) של תת-קטע, את הסכום שהתקבל יהיה פחות מסכום של כל איבריו הקטע. אמנם, אם נתבונן בקטע $a_i, \dots a_{j-1}, a_j, \dots, a_k$ ונתחים לסכום מ-a_j, $(j \geq 2)$ הסכום $a_i + \dots + a_{j-1} > 0$ חיובי , לכן

$$a_j + \dots + a_k < a_i + \dots + a_{j-1} + a_j + \dots + a_k$$

וכאשר מתחילים לסום מאיבר לא מתחילת הקטע מקבלים סכום קטן יותר. אם ממשיכים לסום עם איברי הקטע הבא גם מקבלים סכום קטן יותר בגלל שסכום איברי הקטע הנוכחי הוא שלילי.

מסקנה: בין כל $\frac{n^2}{2}$ קטעים כדאי להתבונן ב-**n** קטעים בלבד. מש"ל.■

הערה חשובה: כאשר כל איברי המערך שליליים הקטע בעל סכום מקסימאלי מורכב מאיבר אחד שהוא האיבר המקסימאלי במערך.

ובכך מתמטיקה אומרת לנו **מה** צרך לעשות ומדעי המחשב אומר **איך** צריך לעשות.

Best pseudo-code

countMax – אורך של תת-מערך בעל סכום מקסימאלי

end – אינדקס+1 של האיבר האחרון של התת-מערך בעל סכום מקסימאלי

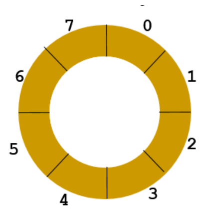
```
best(int[] a)
      i = 0
      while(i<n && a[i]<=0) i++
      if (i==n)
            index = maxIndex(a)
            max = a[index]
            return max
      end-if

      max = a[i],  sum = 0, count = 0
      while(i<n)
            sum = sum + a[i]
            if (sum<=0) sum = 0
            else if (sum>max)
                  max = sum
            end-if
            i++
      end-while
      return max
end-best
```

מציאת קטע בעל סכום איברים גדול ביותר במערך מעגלי.

נזכור את המושג של מערך מעגלי:

מערך מעגלי הוא מערך שהתא האחרון בו "מוצמד" לתא הראשון וכך נוצר מעגל.



שיטת מעבר על מערך מעגלי:

```
for (i=start, k=1; to k≤n;  i=(i+1)%n)
```

חיפוש שלם: עוברים על כל המערכים שנוצרו ממערך מעגלי ונפעיל best על כל אחד מי המערכים האלה:

```
max = a[0]
for i=1 to n
      sum = best(a[i],…,a[n-1], a[0],…,a[i-1])
      if (sum > max) max = sum
end for
```

סיבוכיות של חיפוש שלם היא $O(n^2)$.

הפעלת best על מערך כפול:

```
sum = best(a[0],…,a[n-1], a[0],…,a[n-1])
```

השיטה של מערך כפול לא נותנת תשובה נכונה, נביא דוגמה נגדית:

דוגמה 1: השיטה עובדת בצורה תקינה:

a[] = **1,2**,-100,**5**; max = 5+1+2=8
a2[] = 1,2,-100,**5,1,2**,-100,5; max = 5+1+2=8

דוגמה 2: השיטה נותנת תשובה שגויה :

a[] = **1,2**,-1,**5**; max = 5+1+2=8
a2[] = 1,2,-1,5,1,2,-1,5; max = 14

שיטה המסתמכת על best

יש שתי אפשרויות לקבל קטע בעל סכום מקסימאלי של מערך מעגלי :
א) הקטע נמצא בתוך המערך: $a[i],..,a[j], 0 \le i < j \le \mathrm{n}-1$, לדוגמה:
a[] = 1,2,-100,**5,1,2**,-7
ב) הקטע נמצא על המעגל: $a[j], \dots, a[n-1], a[0], \dots, a[i], i < j$

a[] = **1,2**,-1,**5**

נשים לב לשתי תכונות הבאות של מערך מעגלי:

א)אם קטע $a[i],..,[j]$ הוא בעל סכום מקסימאלי ,כלומר
$sumMax$=$[i]+\ldots+a[j]$ אז סכום של שאר איברי המערך הוא מינימאלי:
$sumMin$=$[j+1]+\ldots+a[n-1]+a[0]+ \ldots+a[i-1]$.

ב) אם קטע $a[i],..,a[j]$ הוא בעל סכום **מקסימאלי** במערך$\boldsymbol{a}$, אז הקטע הזה הוא בעל סכום **מינימאלי** במערך $\boldsymbol{a}$ –.

אלגוריתם:

נסמן ב-sum סכום איברי המערך $sum = a[0]+\ldots+a[n]$.

1. מפעילים **best** על המערך המקורי: `sum1 = best(a)`
2. מפעילים **best** הל המערך a– : `sumNeg = best(-a)` – קבלנו סכום מקסימאלי של המערך השלילי – הסכום הזה הוא סכום מינימאלי במערך מקורי.
   הסכום של שאר איברי המרך a– הוא מקסימאלי במערך המקורי, את הסכום הזה ניתן לחשב לפי הנוסחה:
   `sum2 = sum – (-sumNeg) = sum + sumNeg` .
3. התשובה היא: `sumMax = max(sum1, sum2)`

דוגמה: {10,2,-5,8,-20,12} = []a, {-12,**20**,-8,5 ,-2,-10 } = []a-
sum = 7, sum1 = 15, sumNeg = 20, sum2 = 20+7=27,
sumMax=max(15,27) = 27. .

הערה חשובה: פונקציה **best** מחזירה מספר שלילי אך ורק אם כל איברי המערך הם שליליים. במקרה זה תשובה עבור מערך מעגלי מתקבלת בשלב ראשון של האלגוריתם.

פסדו-קוד:

```
int bestCycle([]a)
   sum1 = best(a)
   if (sum1 < 0) return sum1
   sum = 0
   for i=0 to n-1 sum = sum + a[i]
   t[n] // help array
   for i=0 to n-1 t[i] = -a[i]
   sumNeg = best(t)
   sum2 = sum + sum1
   return maximum(sum1, sum2)
end-bestCycle
```