**אלגוריתם אופטימלי למציאת תת-מערך עם סכום מקסימלי –**

**Best Max-Sum Subinterval of the Interval**

**Algorithm with complexity O(n) - best**

**arr[] = {9,2,-5,-10,-3,50,3,-80,10,20,30,42,-110,30}**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| **a[]** | **9** | **2** | **-5** | **-10** | **-3** | **50** | **3** | **-80** | **10** | **20** | **30** | **42** | **-110** | **30** |
| **sum** | **9** | **11** | **6** | **-4**(0) | **-3**(0) | **50** | **53** | **-23**(0) | **10** | **30** | **60** | **102** | **-8**(0) | **30** |
| **maxSum** |  | **11** |  |  |  |  | **53** |  |  |  |  | **102** |  | **30** |

**best(int[] a)**

**int sum=0, maxSum=a[0]**

**loop from i = 0 to i = a.length – 1 step 1**

**sum = sum + a[i]**

**if (sum < 0)**

**sum = 0**

**else**

**if (sum > maxSum)**

**maxSum = sum**

**loop end**

**return maxSum**

**end-best**

**Result:**

**max = 102**

1. מתחילים לחבר את איברי המערך מאיבר חיובי ראשון.
2. מחברים את האיברים ושומרים על הסכום המקסימאלי עד שנקבל סכום שלילי.

כלומר בכל תת-קטע הסכומים לכל .

1. מאפסים את הסכום וחוזרים לשלב 2.

הערה: לאפיין מקרה כאשר כל האיברים במערך בערך שלילי.

**אלגוריתם אופטימלי למציאת תת-מערך עם סכום מינימלי –**

**Best Min-Sum Subinterval of the Interval**

**Algorithm with complexity O(n) – best**

**A [] = {5, 100, -150, 5, 20, 100}**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **index** | **0** | **1** | **2** | **3** | **4** | **5** |
| **a[]** | **5** | **100** | **-150** | **5** | **20** | **100** |
| **sum** | **5** | **105** | **-45**(0) | **5** | **25** | **125** |
| **maxSum** |  | **105** |  |  |  | **125** |

**Change the signs of all elements of the array on the opposite**:

**-A[] = {-5, -100, 150,-5,-20,-100}**

**AntiDiam(A) = -Diam(-A)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **index** | **0** | **1** | **2** | **3** | **4** | **5** |
| **a[]** | **5** | **100** | **-150** | **5** | **20** | **100** |
| **sum** | **5** | **105** | **-45**(0) | **5** | **25** | **125** |
| **maxSum** |  | **105** |  |  |  | **125** |
| **-a[]** | **-5** | **-100** | **150** | **-5** | **-20** | **-100** |
| **sum** | **-5(0)** | **-100(0)** | **150** | **145** | **125** | **25** |
| **maxSum** |  |  | **150** |  |  |  |
| **minSum** |  |  | **-150** |  |  |  |

**מציאת תת-מערך עם סכום מקסימלי במערך מעגלי**

**Max-Sum subinterval of a Cycle (Circular array)**

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

שיטת המעבר על מערך מעגלי:

**for (i=start, k=1; to k<=n; i=(i+1)%n)**

The **Best Cycle** is the task todetermine the **largest contiguous sum** in a cycle array int[]a (a[i] - positive or negative)

**arr[] = {5, 100, -150, 5, 20, 100}**

**5**

**start**

In the original array, maxSum = 125

**100**

**100**

**-150**

**20**

**5**

1. **חיפוש שלם במטריצה:** 
   1. **בסיבוכיות**

רעיון האלגוריתם: עוברים על כל תתי המערכים/תתי הקטעים שנוצרו מהמערך המעגלי ומפעילים best על כל אחד מתתי המערכים האלו.

כמות התתי הקטעים במערך המעגלי של n איברים היא **:**

תתי הקטעים שמתחילים מאיבר ה1-: [1,1]…[[1,n]

תתי הקטעים שמתחילים מאיבר ה2-: [2,2]…[2,n][[2,1]

תתי הקטעים שמתחילים מאיבר ה-i: [i,i]…[i,n][[i,1]…[i,i-1]

תתי הקטעים שמתחילים מאיבר הn-: [n,n][n,1]…[n,n-1]

היות ועוברים על כל איברי המערך אז הסיבוכיות הכללית היא

max =

for i=1 to n

for j=i to n

s = Best()

if (s > max)

max = s

end for

end for

* 1. **משופר בסיבוכיות**

max =

for i=1 to n

s = Best()

if (s > max)

max = s

end for

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6(0)** | **7(1)** | **8(2)** | **9(3)** | **10(4)** | **best** |
| **5** | **100** | **-150** | **5** | **20** | **100** |  |  |  |  |  | **125** |
|  | **100** | **-150** | **5** | **20** | **100** | **5** |  |  |  |  | **130** |
|  |  | **-150** | **5** | **20** | **100** | **5** | **100** |  |  |  | **230** |
|  |  |  | **5** | **20** | **100** | **5** | **100** | **-150** |  |  | **230** |
|  |  |  |  | **20** | **100** | **5** | **100** | **-150** | **5** |  | **225** |
|  |  |  |  |  | **100** | **5** | **100** | **-150** | **5** | **20** | **205** |
|  |  |  |  |  |  |  |  |  |  |  |  |

maxSumCycle = 230

1. **חיפוש שלם משופר באמצעות תכנון דינאמי במטריצה - בסיבוכיות**

**משולש עליון** **במטריצה** - תא שמאופיין ע"י (i,j) כך ש-i<j מהוה תא סטנדרטי במערך רגיל שמחושב לפי התכנון הדינאמי באופן הבא:

**mat[i][j] = mat[i+1][j] + mat[i][i]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |  |
| 5 | 105 | -45 | -40 | -20 | -70 | 0 |
| 80 | 100 | -50 | -45 | -25 | -75 | 1 |
| -20 | 80 | -150 | -145 | -125 | -25 | 2 |
| 130 | **230** | 80 | 5 | 25 | **125** | 3 |
| 125 | 225 | 75 | 80 | 20 | 120 | 4 |
| 105 | 205 | 55 | 205 | 80 | 100 | 5 |

**משולש תחתון במטריצה** – כל תא המשולש התחתון שמאופיין ע"י [i,j] כך ש-i>j מהווה איבר במערך מעגלי כלומר כל תא המשולש התחתון הוא תתי המקטעים הרצפים בצורה מעגלית.

איך לחשב אותו?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** |
| **5** | **100** | **-150** | **5** | **20** | **100** |

לדוגמה, נחשב סכום המקטע [5,1] כלומר המקטע מכיל איברים במיקומים 5, 0, 1.

אם נשתמש בתבונה כי סכום המקטע כלשהו במערך מעגלי זה סכום של כל האיברים במערך פחות סכום המקטע שאר האיברים. אז סכום המקטע [5,1] יהיה שווה

סכום של כל האיברים [0,5]

פחות סכום המקטע של שאר האיברים [2,4]:

-

מקרה פרטי, כאשר [i,j] כך ש-i>j ו-i=j+1 כוונה j,i רצפים, המקטע [i,j] הוא מעגל בגודל המערך הרגיל וסכום המקטע שווה לסכום כל איברים במערך.

mat[5][0]=80-mat[1,4]=80-(-25)=105 mat[5][1]=80-mat[2,4]=80-(-125)=205

mat[4][0]=80-mat[1,3]=80-(-45)=125 mat[4][1]=80-mat[2,3]=80-(-145)=225

mat[3][0]=80-mat[1,2]=80-(-50)=130 mat[3][1]=80-mat[2,2]=80-(-150)=230

mat[2][0]=80-mat[1,1]=80-100=-20 **mat[2][1]=80-mat[2,1]=80-0=80**

**mat[1][0]=80-mat[1,0]=80-0=80**

mat[5][2]=80-mat[3,4]=80-25=55 mat[5][3]=80-mat[4,4]=80-(-125)=205

mat[4][2]=80-mat[3,3]=80-5=75 **mat[4][3]=80-mat[4,3]= 80-0=80**

**mat[3][2]=80-mat[3,2]= 80-0=80**

maxSumCycle = 230

1. **בעזרת מערך כפול - בסיבוכיות**

רעיון האלגוריתם: הפעלת best על מערך כפול:

Doubled array + Best()

arr[] = {5, 100, -150, 5, 20, 100}

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doubled array | 5 | 100 | -150 | 5 | 20 | 100 | 5 | 100 | -150 | 5 | 20 | 100 |
| sum | 5 | 105 | -45(0) | 5 | 25 | 125 | 130 | 230 | 80 | 85 | 105 | 205 |
| maxSum |  | 105 |  |  |  | 125 | 130 | 230 |  |  |  |  |

maxSumCycle = 230

arr[] = {1, 2, -1 , 5}

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doubled array | 1 | 2 | -1 | 5 | 1 | 2 | -1 | 5 |
| sum | 1 | 3 | 2 | 7 | 8 | 10 | 9 | 14 |
| maxSum |  |  |  |  |  |  |  | 14 |

maxSumCycle = 14

אך הסכום הגדול ביותר בתת המקטע במערך המעגלי הוא 8.

אז השיטה של המערך הכפול לא תמיד נותנת תשובה נכונה בסיבוכיות

על מנת לשפר את השיטה, נצטרך לעבור על כל תתי הקטעים בגודל המערך שקיימים במערך המעגלי ועל כל תת הקטע להפעיל Best אז נקבל שוב את הסיבוכיות .

1. **בעזרת המינוס של המערך - בסיבוכיות**

שיטת הסימטריה המסתמכת עלbest -Best-Cycle(A) .

יש שתי אפשרויות לקבל מקטע בעל סכום מקסימאלי של מערך מעגלי :

* המקטע נמצא בתוך המערך הרגיל:
  + לדוגמה: A[] = {1,2,-100,5,1,2,-7}
* המקטע נמצא על המעגל:
  + לדוגמה: A[] = {1,2,-1,5}

נשים לב לשתי תכונות הבאות:

1. אם המקטע 𝑎[𝑖], . . , 𝑎[𝑗] בעל סכום מקסימאלי, כלומר , אז סכום של שאר איברי המערך הוא מינימאלי:

1. אם המקטע 𝑎[𝑖], . . , 𝑎[𝑗] בעל סכום מקסימאלי במערך A, אז למשלים המערך –A המקטע 𝑎[𝑖], . . , 𝑎[𝑗] הוא בעל סכום מינימאלי.

**אלגוריתם:**

1. מחשבים סכום איברי המערך המקורי:
2. נפעיל Best על המערך המקורי (הרגיל):
3. נפעיל Best על המערך המשלים: קיבלנו סכום מקסימלי במערך המשלים שהוא סכום מינימלי במערך המקורי (לפי התכונה 2).

לפי התכונה 1, סכום של שאר איברי המערך *הוא סכום מקסימלי במערך המקורי, את הסכום ניתן לחשב לפי הנוסחה הבאה:*

1. *על מנת למצוא את הסכום המקסימלי במערך המעגלי ניקח את הערך המקסימלי בין תוצאות החישוב בסעיפים 2 ו-3:*

דוגמה:

A[] = {5, 100, -150, 5, 20, 100}

[] = {-5, -100, 150, -5, -20, -100}

(125,230)=230

**Pseudo-code:**

**Best-Cycle**(int[] a)

int sum1 = best(a)

int t[] = new int[a.length]

int sumA = 0

for (int i = 0; i < t.length; i++)

sumA = sumA + a[i]

t[i] = -a[i]

end-for

int sum2 = sumA + best(t)

if (sum1 >= sum2)

return sum1

else

return sum2

**end-Best-Cycle**