

ש.ב. שרשרת חוליות – אופיר הופמן י"א 3

1. CompareList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 3
7-->4-->0-->1-->5-->
7-->4-->0-->1-->5-->
TEST 1: OK
7-->4-->0-->1-->5-->
7-->4-->0-->1-->
TEST 2: OK
7-->4-->0-->1-->5-->
1-->2-->3-->4-->
TEST 3: OK
_
```

סיבוכיות: $O(n)$ (n אורך הרשימות) – התוכנית עוברת על שתי הרשימות במקביל פעם אחת ומשווה את האברים שלהן

2. CountList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 5
1-->2-->3-->4-->5-->
result1=5
2-->5-->8-->
result2=3
1-->6-->4-->7-->2-->3-->4-->8-->
result3=8
```

סיבוכיות: $O(n)$ (n אורך הרשימה) מפני שהתוכנית עוברת על הרשימה פעם אחת

3. SumList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 7
1-->2-->3-->4-->5-->
result1=15
6-->0-->1-->
result1=7
1-->0-->0-->0-->0-->0-->1-->
result1=2
```

סיבוכיות: $O(n)$ (n אורך הרשימה). כי התוכנית עוברת על כל האיברים פעם אחת וסוכמת.

4. IsExist

```
C:\WINDOWS\system32\cmd.exe
enter test number: 8
1-->2-->3-->4-->5-->
val=3 : True
val=1 : True
val=6 : False
```

סיבוכיות (במקרה הגרוע): $O(n)$ (n אורך הרשימה). כי התוכנית עוברת על כל האיברים פעם אחת ומשווה לvalue.

5. FindMax

```
C:\WINDOWS\system32\cmd.exe
enter test number: 9
1-->2-->3-->4-->5-->
Max=5
-1-->0-->-4-->
Max=0
99-->
Max=99

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה). כי התוכנית עוברת על כל האיברים פעם אחת ומשווה ל max הנוכחי.

6. AbsValue

```
C:\WINDOWS\system32\cmd.exe
enter test number: 10
1-->-2-->3-->-4-->-5-->
Result :
1-->2-->3-->4-->5-->

-1-->0-->-4-->
Result :
1-->0-->4-->

1-->2-->3-->4-->-99-->
Result :
1-->2-->3-->4-->99-->

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה). כי התוכנית עוברת על כל האיברים פעם אחת ועושה ערך מוחלט.

7. GetNodeRef

```
C:\WINDOWS\system32\cmd.exe
enter test number: 11
1-->2-->3-->4-->5-->
location=3, value=3

1-->2-->3-->10-->
location=4, value=10

1-->2-->3-->9-->
location=5, value=
_
```

סיבוכיות: $O(k)$ (k מיקום האיבר ברשימה). כי התוכנית עוברת על כל איבר פעם אחת עד האיבר ה-k.

8. IsSorted

```
C:\WINDOWS\system32\cmd.exe
enter test number: 12
0-->1-->2-->3-->4-->5-->6-->7-->8-->9-->
sorted=True

2-->2-->3-->9-->
sorted=True

1-->2-->4-->5-->8-->8-->7-->
sorted=False

10-->20-->30-->19-->40-->50-->
sorted=False

2-->
sorted=True
_
```

סיבוכיות: $O(n)$ (n אורך הרשימה) מפני מעבר יחיד על כל איברי הרשימה והשוואה לאיבר הבא.

9. Count Sequences

```
C:\WINDOWS\system32\cmd.exe
enter test number: 13

4-->5-->6-->6-->3-->4-->4-->4-->22-->3-->
value=4, result=2

7-->3-->7-->7-->7-->9-->0-->-1-->7-->7-->3-->3-->
value=7, result=3

1-->2-->3-->
value=4, result=0
```

סיבוכיות: $O(n)$ (n אורך הרשימה). התוכנית עוברת על כל איברי הרשימה ומשווה לערך שהתקבל כפרמטר. וסופרת כמה רצפים יש לערך

10. PrintAtoB

```
C:\WINDOWS\system32\cmd.exe
enter test number: 14

1-->2-->3-->4-->5-->6-->7-->8-->9-->
input=5,8 | result=5-->6-->7-->8-->

3-->8-->6-->7-->7-->9-->4-->5-->3-->1-->
input=6,9 | result=9-->4-->5-->3-->

110-->32-->11-->76-->
input=1,4 | result=110-->32-->11-->76-->
```

סיבוכיות: $O(n)$ (n אורך הרשימה, במקרה הגרוע ביותר). התוכנית "מדלגת" למיקום הראשון ועוברת על האיברים ומדפיסה אותם עד למיקום השני.

11. RemoveDuplicates

```
C:\WINDOWS\system32\cmd.exe
enter test number: 17

5-->3-->3-->2-->0-->1-->
result=5-->3-->2-->0-->1-->

2-->6-->4-->2-->8-->4-->4-->6-->1-->9-->
result=2-->6-->4-->8-->1-->9-->

1-->2-->3-->4-->5-->1-->5-->5-->
result=1-->2-->3-->4-->5-->

Press any key to continue . . .
```

סיבוכיות: $O(n^2)$ (n אורך הרשימה המקורית). עובר על הרשימה, עבור כל חוליה עובר על הרשימה החדשה לבדיקה של כפילות.

12. BalancedList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 15
5-->10-->2-->8-->7-->3-->
(expecting true)=>True

2-->9-->5-->7-->1-->6-->3-->8-->
(expecting true)=>True

10-->20-->30-->40-->50-->60-->70-->
(expecting true)=>True

1-->2-->3-->4-->5-->10-->
(expecting false)=>False
```

סיבוכיות: $O(2n)$ (n אורך הרשימה). התוכנית עוברת על הרשימה פעם אחת לבדיקה ממוצע ופעם נוספת לבדיקה האם הרשימה מאוזנת.

13. RemoveMax

```
Select C:\WINDOWS\system32\cmd.exe
enter test number: 16
5-->7-->2-->8-->10-->3-->
max value = 10
Updated list : 5-->7-->2-->8-->3-->

2-->9-->5-->7-->1-->6-->3-->8-->
max value = 9
Updated list : 2-->5-->7-->1-->6-->3-->8-->

7-->3-->7-->7-->7-->9-->0-->-1-->7-->7-->3-->3-->
max value = 9
Updated list : 7-->3-->7-->7-->7-->0-->-1-->7-->7-->3-->3-->
```

סיבוכיות: $O(n)$ (n אורך הרשימה) . התוכנית עוברת על איברי הרשימה ומוצאת את החוליה עם הערך המקסימלי והחוליה הקודמת לה, מנתקת את המקסימלית ומחברת את הקודמת להבאה.

14. InsertToSortedList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 18

1-->2-->3-->5-->
new value = 4 | result : 1-->2-->3-->4-->5-->

1-->
new value = 2 | result : 1-->2-->

1-->2-->2-->19-->23-->24-->
new value = 7 | result : 1-->2-->2-->7-->19-->23-->24-->

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה) מפני שהתוכנית עוברת על הרשימה עד למציאת המיקום של החוליה החדשה ברשימה הממויינת.

15. Sort

```
C:\WINDOWS\system32\cmd.exe
enter test number: 19

3-->2-->1-->5-->4-->
Sorted : 1-->2-->3-->4-->5-->

5-->2-->3-->1-->3-->9-->10-->7-->4-->
Sorted : 1-->2-->3-->3-->4-->5-->7-->9-->10-->

100-->90-->80-->70-->65-->35-->11-->78-->-2-->0-->
Sorted : -2-->0-->11-->35-->65-->70-->78-->80-->90-->100-->

Press any key to continue . . .
```

סיבוכיות: $O(n^2)$ (n אורך הרשימה). עבור כל חוליה ברשימה התוכנית עוברת על כל הרשימה חדשה הממויינת למציאת המיקום של החוליה (מיון הכנסה, שימוש בinsertToSortedList).

16. ReverseList

```
C:\WINDOWS\system32\cmd.exe
enter test number: 2
4-->8-->2-->5-->8-->
reversed : 8-->5-->2-->8-->4-->

5-->2-->3-->1-->3-->9-->10-->7-->4-->
reversed : 4-->7-->10-->9-->3-->1-->3-->2-->5-->

4-->7-->2-->9-->5-->
reversed : 5-->9-->2-->7-->4-->

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה). עובר על הרשימה פעם אחת ומשנה את כיוון הקישורים.

17. NegativeSequence

```
C:\WINDOWS\system32\cmd.exe
enter test number: 21
2-->-1-->-3-->5-->
result=2

4-->-2-->-9-->1-->2-->-8-->-3-->-1-->6-->-5-->-1-->
result=3

-5-->1-->1-->-9-->3-->-8-->-4-->-99-->-12-->0-->0-->-5-->-11-->-5-->100-->-23-->
result=4

1-->2-->3-->7-->3-->0-->
result=0

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה), התוכנית עוברת על כל חוליה ברשימה, סופרת רצפים של מספרים שליליים ומעדכנת את אורך הרצף הגדול ביותר במידה ומצאה אחד.

18. MergeSortedLists

```
C:\WINDOWS\system32\cmd.exe
enter test number: 21

list1 : 1-->4-->9-->
list2 : 2-->5-->
merged :
1-->2-->4-->5-->9-->

1-->5-->9-->22-->
0-->2-->11-->24-->220-->
merged :
0-->1-->2-->5-->9-->11-->22-->24-->220-->

10-->20-->30-->40-->50-->100-->
1-->2-->3-->4-->19-->20-->40-->42-->99-->101-->
merged :
1-->2-->3-->4-->10-->19-->20-->20-->30-->40-->40-->42-->50-->99-->100-->101-->

Press any key to continue . . .
```

סיבוכיות: $O(n+k)$ (n, k אורכי הרשימות). התוכנית משווה בין שני הערכים הראשונים, מכניסה את הערך הקטן לרשימה החדשה ועוברת לחוליה הבא של אותה רשימה. לבסוף מכניסה את ה"זנב" של השרשרת שלא כל הערכים שלה הוכנסו.

19. MergeSortedLists 2

```
C:\WINDOWS\system32\cmd.exe
enter test number: 21

list1 : 1-->4-->9-->
list2 : 2-->5-->
merged :
1-->2-->4-->5-->9-->

2-->3-->9-->
1-->1-->
merged :
1-->2-->3-->9-->

10-->20-->30-->40-->50-->100-->
1-->2-->3-->4-->19-->20-->40-->42-->99-->101-->
merged :
1-->10-->19-->20-->20-->30-->40-->50-->99-->100-->101-->

Press any key to continue . . .
```

סיבוכיות: כמו הפתרון הקודם (MergeSortedLists)

20. AddFollowingNum

```
C:\WINDOWS\system32\cmd.exe
enter test number: 23

9-->15-->20-->
val=15 | result : 9-->15-->16-->20-->

1-->3-->7-->11-->
val=7 | result : 1-->3-->7-->8-->11-->

-3-->2-->18-->9-->
val=9 | result : -3-->2-->18-->9-->10-->

Press any key to continue . . .
```

סיבוכיות: $O(n)$ (n אורך הרשימה). התוכנית עוברת על הרשימה עד למציאת הערך ואז מוסיפה חוליה חדשה עם הערך.

21. ListsIntersections

```
C:\WINDOWS\system32\cmd.exe
enter test number: 24

3-->1-->5-->
2-->3-->6-->
result : 3-->

1-->8-->2-->6-->3-->
9-->1-->7-->3-->6-->
result : 1-->6-->3-->

4-->-2-->18-->55-->27-->
-2-->0-->1-->4-->
result : 4-->-2-->

Press any key to continue . . .
```

סיבוכיות: $O(n \cdot k)$ (n אורכי הרשימות). עבור כל חוליה ברשימה 1 התוכנית בודקת האם הערך קיים ברשימה 2 (באמצעות IsExist).