**שאלה 1 – חסמי סיבוכיות (25 נק')**

**חלק א':** **חימום**

1. **(2 נק')** אלו מהפונקציות הבאות הינן? אלו מהפונקציות הבאות הינן ?

**אין צורך** בהסברים או הוכחות בסעיף זה, אלא רק רשימה של פונקציות לכל מקרה, אך יתכן ותרצו להוכיח זאת לעצמכם, על מנת לוודא צדקתכם.

2. ,




8. **(1 נק')** הוכיחו **פורמלית** ש-

**הוכחה:**

נניח בשלילה כי מתקיים ונגיע לסתירה.

אם ההנחה נכונה אזי קיים  *כך שלכל מתקיים :*

*עבור הטענה לא מתקיימת, ולכן .*

**חלק ב':** **חקר פונקציות**

לפניכם 15 פונקציות:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | |

**(22 נק')** נסמן אם מתקיים . סדרו את הפונקציות לפי סדר אסימפטוטי.

**מצאו קבועים מתאימים לפי הגדרה**, הוכיחו את תשובתכם, ואת כל החישובים עליהם אתם מתבססים. נוסף על כך, אם שתי פונקציות בסידור מקיימות גם

, ציינו זאת והוכיחו.

**פתרון:**

עבור כל פונקציה ננסה תחילה לפשט ככל הניתן את פונקציית הסיבוכיות שלה, ולתת פונקציה נוחה יותר לחישובים בהמשך.

* **עבור** **:**

מקורס אינפי 1, הוכחנו באינדוקציה את המשוואות הבאות:



לגבי משוואה (1) נחזור ונוכיח אותה באינדוקציה (ליתר ביטחון),

בסיס האינדוקציה – עבור  נקבל:



צעד האינדוקציה – עבור  הטענה נכונה.

צריך להוכיח כי הטענה עבור  נכונה, כלומר צריך להוכיח כי



הוכחה --



כנדרש !

נפתח כעת את :



קיבלנו כאן פולינום.

נחשב כעת את המקדם של החזקה הגבוהה ביותר (( ונקבל:



מכאן נסיק כי .

* **עבור :**

*טענות:*

1. ***-log= a\*log(x)***

* **עבור :**

נחסום את הפונקציה רק מצד אחד



בנוסף, כל פונקציה היא  ולכן .

* **עבור :**

*טענות:*

1. **סכום סדרה הנדסית עם q=2 ו-n איברים**

* **עבור :**

ידוע כי (נוסחת סטרלינג) ולכן,



באותו אופן,



נפתח כעת את :



מצד אחד:



ומצד שני:



ולכן, נבחר  ו- ואז  מתקיים ש:



ולכן 

* **עבור :**

*טענות:*

1. **=**

* **עבור :**

נעזר בנוסחה



נפתח כעת את :



נחסום את הביטוי שקיבלנו.



נבחר ואז מתקיים:



ולכן .

* **עבור :**
* **עבור :**

*טענות:*

* **עבור :**

נחסום את הביטוי משני הצדדים.

מצד אחד,



ומצד שני,



נבחר ואז מתקיים:



ולכן .

* **עבור :**

*טענות:*

1. **סכום סדרה חשבונית + סכום בינום**
2. **החל מ-**

* **עבור :**

נחסום את הביטוי משני הצדדים:

מצד אחד,



מצד שני, אם נסתכל על הפונקציה נראה כי בכל אינטרוול , , הפונקציה חוסמת מלבן שצלעותיו  ושטחו  מה שגורר כי , ולכן:



נבחר  ואז מתקיים



ולכן .

* **עבור :**

*טענות:*

1. **Log(X\*Y) = log(X)\*log(Y)**

* **עבור :**

ניזכר בבינום של ניוטון



ונציב ונקבל:



סה"כ נקבל כי,



ולכן 

**לסיכום:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**שאלה 2 - רקורסיה (20 נק')**

1. הוכיחו באינדוקציה כי: .

**פתרון:**

נוכיח באינדוקציה כי לכל  מתקיים, .

אם נוכיח זאת, נבחר  ו- , ואז לכל  מתקיים:



ולכן  כנדרש.

נוכיח את הטענה:

בסיס האינדוקציה – נבדוק את הטענה עבור :

עבור 



עבור 



צעד האינדוקציה – נניח כי הטענה נכונה לכל  ונוכיח כי  .

הוכחה:



כנדרש !

1. נתונה משוואת הנסיגה הבאה: .

כאשר a הוא קבוע ו- לכל האם קיים a קבוע עבורו ? אם כן מהו ה-a המינימאלי עבורו זה מתקיים ומה ערכה האסימפטוטי של במקרה זה?

פתרון:

נבצע תחילה החלפת משתנים-

נחלק את שני האגפים ב-



נציב . נשים לב כי תחת הגדרה זו מתקיים:





סה"כ נקבל את נוסחת הנסיגה הבאה: .

בשאלה דורשים עבור אילו ערכי a 

לפי ההגדרה, זה אומר שעבור אילו ערכי a, לכל  קיים  כך שלכל :

. וזוהי בדיוק ההגדרה לכך ש-.

נפתח את  בעזרת עצי רקורסיות: (עמוד הבא)









































|  |
| --- |
|  |
|  |
|  |

נסמן  , ונקבל כי  . מסנדוויץ' נקבל כי



כדי שהגבול ישאף לאינסוף (סכום סדרה הנדסית), יש צורך כי , כלומר 

תשובה סופית: a המינימאלי עבורו  הוא a=12.

1. מצאו חסם עליון וחסם תחתון אסימפטוטיים פשוטים (כלומר, ללא שימוש בסכימה, סימן עצרת וכו') עבור בכל אחת מנוסחאות הנסיגה שלהלן. מצאו חסמים הדוקים ככל שתוכלו והוכיחו תשובתכם.

פתרון:

1. נסמן  ,, .

ואז  . נבדוק את תנאי משפט המאסטר:

נוכיח כי 

הוכחה:

נעזר באי שוויון  לכל n, ונחסום את  .

מצד אחד-



מצד שני-



נבחר   ואז לכל  מתקיים:



ולכן 

ולפי משפט המאסטר 🡨 

1. נסמן  ,, 

ואז  . נבדוק את תנאי משפט המאסטר:

נבחר  נוכיח כי 

הוכחה:



ולפי משפט המאסטר 🡨 

1. נפתור את הבעיה בשלבים:

*שלב ראשון* – נמצא חסם אסימפטוטי לנוסחת הנסיגה הבאה:



פתרון –

נסמן  ,, 

ואז  . נבדוק את תנאי משפט המאסטר:

קל לראות כי , ולכן

לפי משפט המאסטר 🡨 

*שלב שני* – נמצא חסם אסימפטוטי לנוסחת הנסיגה הבאה:



צעד 1: נסמן .



צעד 2: נגדיר . נשים לב כי תחת הגדרה זו, 

ואז 

*בשלב הראשון* ראינו ש-  ולכן:



*שלב שלישי* – נפתור את הבעיה שלנו:



תחילה נחלק את המשוואה ב-:



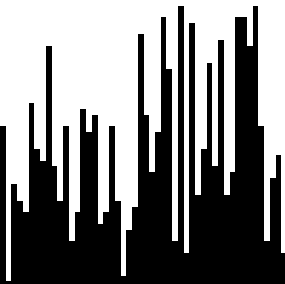
נציב . נשים לב כי תחת הגדרה זו, 

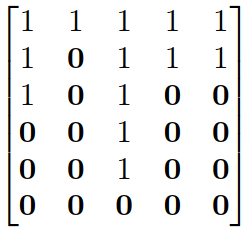
ואז 

*בשלב שני* ראינו ש-  ולכן:



**שאלה 3 – חקר אסימפטוטי של שני משתנים (20 נק')**

נתונה לכם תמונת בינארית של קו גורדי השחקים בתל אביב. התמונה היא בגובה פיקסלים, וברוחב פיקסלים, ובתמונה בניינים צבועים שחור (ערך פיקסל 0) ושמיים צבועים לבן (ערך פיקסל 1). בכל טור, כל הפיקסלים השחורים נמצאים תמיד מתחת לפיקסלים הלבנים. דוגמה להמחשה עבור תמונה של (ימין) ותצוגה בינארית של תמונה בגודל (שמאל):



בבעיה זו, תתכננו אלגוריתם יעיל למציאת **הבניין הגבוה ביותר** בתמונה.

**קלט:** מטריצה בגודל , בה הבניינים מסומנים באפסים, והשמיים באחדות.

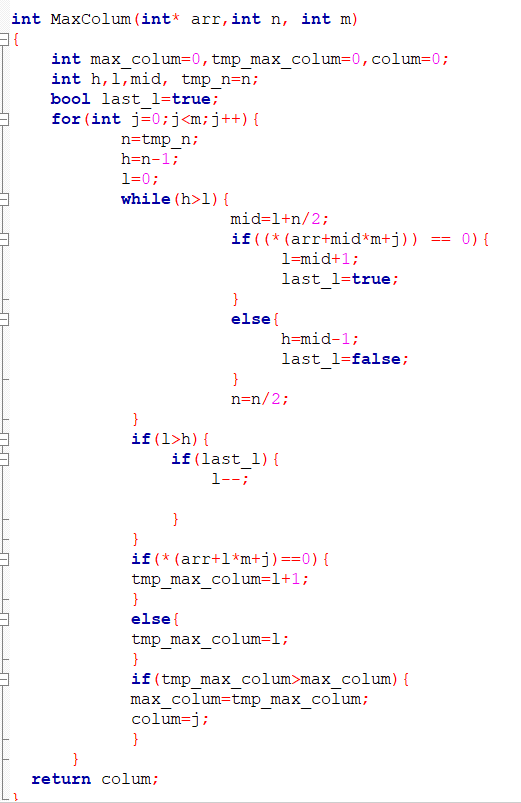
**פלט:** אינדקס הטור בו שוכן הבניין הגבוה ביותר. למשל, בדוגמה השמאלית מעלה, הבניין הגבוה ביותר הינו בגובה , ונמצא באינדקס , ועל כן הפלט יהיה 1.

1. **(5 נק')** מצא אלגוריתם הפותר את הבעיה בזמן .

**פתרון:**

נבצע חיפוש בינארי על כל עמודה נחפש את המקום בו יש 0 הכי גבוה (המופע האחרון של 0). חיפוש בינארי על עמודה באורך n עולה log(n) ויש m עמודות לכן mlog(n).

נשמור את הערך עמודה המקסימאלית במשתנה ואת האינדקס המתאים לה במשתנה נוסף.



נתחיל מאמצע עמודה הראשונה אם היא 0 אז low=mid אם זה 1 אז high= mid ונמשיך עם מערך בגודל n/2 וכן הלאה עד ש high<=low כמו בכל חיפוש בינארי.

נישמור את גודל הבניין במשתנה max\_colum ואת מספר העמודה המתאימה במשתנה colum אם נגיע לבניין גבוה יותר אז יוחלף לערך הגדול וכן J לעמודה המתאימה.

נחזיר את colum .

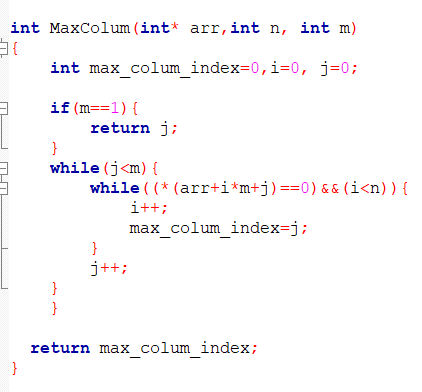
1. **(5 נק')** מצא אלגוריתם הפותר את הבעיה בזמן .

**פתרון:**

1)נתחיל מתחתית העמודה הראשונה

2)נתפס לאורך הבניין (אפסים) עד שנתקל ב-1 נשמור את העמודה במשתנה ונמשיך באותה שורה לעמודה הבאה

3) נחזור על- 2 עד לעמודה האחרונה (m). בסה"כ יש לנו הליכה על סריג NxM עם צעדים למעלה או ימינה לכן סה"כ n+m צעדים. נחזיר את המשתנה של העמודה האחרונה ששמרנו.



[ עבור סעיפים א' וב' – אנו מצפים לפסאודו קוד, הסבר **ברור** בעברית על האלגוריתם והצדקה קצרה על סיבוכיות הזמנים. אין צורך בהוכחת נכונות של האלגוריתם ]

1. **(3 נק')** עבור ערכים מסוימים של האלגוריתם מחלק א' יעיל יותר, ואולם, לערכים אחרים, האלגוריתם מחלק ב' עדיף. לכל ערך של במובנים של מטה, הסבירו איזה אחד מהאלגוריתמים יעיל יותר (או אולי, הם יעילים באותה המידה?) באופן אסימפטוטי. המקרה עבורו מולא עבורכם כדוגמה. אנא ספקו את הביטוי האסימפטוטי באופן **הפשוט ביותר** שניתן.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | זמן ריצה עבור א' |
|  | O( |  |  |  |  |  | זמן ריצה עבור ב' |
| א | א | שווים | ב' | ב | ב | שווים | מי עדיף? |

פתרון:

**n=100**

א: m(log(100)) =O(m)

ב: m+100=O(m)

n=

א:mlog(m^0.5)=0.5\*mlogm=O(mlogm)

ב:m+m^0.5=O(m)

א:

ב:

א:

ב:

א:

ב:

**(7 נק')** שלבו את שני האלגוריתמים מחלקים **א' וב'** על מנת ליצור אלגוריתם יחיד משופר. דהיינו, לכל צמד , האלגוריתם החדש אמור להיות אסימפטומטי לא גרוע מ-א' או ב', ובמקרים מסוימים – טוב משניהם. הציגו את האלגוריתם עם הסבר ברור, כתבו עבורו פסאודו-קוד, ספקו את סיבוכיות זמן הריצה שלו, והציגו דוגמה של פונקציה כך שאם , האלגוריתם שלכם **טוב ממש** אסימפטוטית מהמדדים של ו-

**רמז:** הסיקו מסקנות מסעיף ג' – מתי א' טוב יותר? מתי ב' טוב יותר?

פתרון:

כל עוד n>m

נבצע חיפוש בינארי על שורות המערך ז"א :

נתחיל בשורה האמצעית אם יש רק אחדים בכל השורה נוריד את h=mid

אחרת

נשמור במערך בגודל m את כל העמודות בהם יש 0 ו-l=mid

נחלק את n=n/2

ברגע שm>=n

נמשיך עם אלגוריתם ב'

**שאלה 4 – מבני נתונים תחת סיבוכיות ( (15 נק')**

יחידת המעקב של האו"ם רוצה לעקוב אחר מועדי הבחירות במדינות השונות בעולם. ידוע שיש בעולם לכל היותר מדינות דמוקרטיות (בהן נערכות בחירות) המיוצגות ע"י האינדקסים עד . במדינה מתקיימות בחירות באופן מחזורי כל ימים, כפי שמעוגן בחוקי אותה מדינה. למשל, בישראל עוברים 1461 ימים בין ימי בחירות עוקבים. יהי הציעו מבנה נתונים למימוש הפעולות הבאות בסיבוכיות הזמן הרשומה ליד כל פעולה.

הערה: ו- אינם קבועים לצורכי ניתוח סיבוכיות.

אתחול של מבנה נתונים ריק עם הפרמטרים ו- .

סיבוכיות זמן: .

הוספת המדינה למבנה, עם מחזור בחירות בן ימים. מציין את מספר הימים שנותרו עד לבחירות הבאות במדינה מיום ביצוע פעולה זו. מתקיים . הניחו כי המדינה לא התווספה בעבר למבנה.

סיבוכיות זמן: .

הדפסת האינדקסים של כל המדינות שבהן נערכות בחירות ביום הפעלת הפונקציה. הניחו כי מתבצעת קריאה לפונקציה זו בדיוק פעם אחת בתחילת כל יום, כלומר זוהי הפעולה הראשונה על המבנה באותו יום (כלומר ניתן להניח שפעולה זו מסמנת את סוף היום הקודם).

סיבוכיות זמן: כאשר זהו מספר המדינות המודפסות.

הקדמה של מערכת הבחירות הקרובה של מדינה ב – ימים.

סיבוכיות זמן: .

(באו"ם הבחינו שבמדינה קטנה כלשהי במזרח התיכון נוטים להקדים מערכות בחירות באופן תדיר, לכן דרושה פעולה זו (.

סיבוכיות המקום הנדרשת עבור המבנה הינה .

פתרון:

לפני שנתחיל לפתור, נגדיר אובייקט :

– מס' האינדקס של המדינה

– מחזור ימי הבחירות של המדינה

– המיקום של האובייקט במערך שנראה בהמשך

–של מדינות האיבר הבא ברשימה

– האיבר הקודם ברשימה של מדינות

נממש תחילה את המבנה, ואחרי זה נחשב את הסיבוכיות שלו, ונראה את נכונותו.

הפונקציה תקצה מקומות במבנה הנתונים שלנו:

– מס' מחזור הבחירות הגבוהה ביותר

– מערך כתובות למדינות שנבנה

– מערך שכל תא מייצג יום (יעזור לנו בחישובים בהמשך)

– איטרטור אותו נקדם, ויעזור לנו לעבור על המערך 

בנוסף למקומות אלה, נקצה עוד מערכים לצורך אתחול המערכים  ו- ל- לפי השיטה שנלמדה בכיתה.

התאים במערך  יהוו מצביעים להתחלה של רשימה דו כיוונית.

נסיים את ריצת הפונקציה.

בפונקציה נקצה אובייקט .

באובייקט עצמו נשמור את הערכים  שקיבלנו, ובערך  של האובייקט נשמור .

נשמור את הכתובת של האובייקט שיצרנו ב-.

בנוסף, נשרשר את הכתובת של האובייקט לתחילת הרשימה השמורה ב 

נסיים את ריצת הפונקציה.

בפונקציה, כל פעם נבצע את השלבים הבאים:

נגדיל את ה- ב-1 מודולו D, דהיינו .

נרוץ על הרשימה השמורה ב – כאשר כל איבר ברשימה :

נדפיס את האינדקס  שלו, נעדכן את  שלו, ננתק אותו מהרשימה הנוכחית שלו, ונשרשר אותו לתחילת הרשימה הנמצאת ב-.

נסיים את ריצת הפונקציה.

בפונקציה , נבצע את השלבים הבאים:

ניגש לאיבר בתא , נעדכן את , , ננתק אותו מהרשימה הנוכחית שלו, ונשרשר אותו לתחילת הרשימה הנמצאת ב-.

נסיים את ריצת הפונקציה.

הערות:

1. נשים לב שאנו פונים למקומות ממשיים במערכים שלנו. כך למשל כאשר אנו ניגשים ל-, אז מתקיים כי .

מאחר ופעולת מודולו מביאה ערך שהוא קטן מ-D.

1. הפעולה של שרשור לתחילת הרשימה מתבצעת כך: אם הרשימה ריקה, תחילת הרשימה תצביע על האיבר המשורשר, אחרת, האיבר הראשון ברשימה (ניתן לגשת אליו בסיבוכיות זמן של O(1) מאחר ואנו פונים אליו ישירות) יצביע לאיבר ברצוננו לשרשר עם , והאיבר המשורשר יצביע אליו עם , ותחילת הרשימה תצביע על האיבר החדש. סך-הכל מתבצע מס' פעולות סופי ולכן זה חסום ב .

בדיקת נכונות המבנה:

נראה את הרעיון של התנהלות המבנה. ידוע לנו כי כול יום שעובר אנחנו מקדמים את ה- שלנו ב-1 מודולו , וכאשר אנו מוסיפים מדינה חדשה אנחנו מעדכנים את מיקום המדינה במערך  ביחס למיקום של  פלוס המרחק שלו מהיום המיועד לבחירות, כך שלאחר הזמן המיועד נדפיס בדיוק את המדינה שלנו, וכאשר נדפיס את המדינה, נתחיל את המחזור הספירה החדש עוד פעם ביחס ל-.

במקרה ונרצה להקדים את הבחירות, השינוי שיתבצע במערך  יהיה שינוי ציקלי ביחס ל-, וכן יקטין את המרחק בינו לבין  שגם הוא נע בצורה ציקלית.

בדיקת סיבוכיות המבנה:

פונקציה –

* אתחלנו מס' סופי של איברים – סיבוכיות זמן 
* פעולת הקצאת מקום  למערכים  ו- – סיבוכיות זמן 
* אתחול מערך לפי האלגוריתם שלמדנו בכיתה – סיבוכיות זמן  וסיבוכיות מקום  עבור המערך  ו- עבור מערך , סה"כ סיבוכיות מקום .

פונקציה –

* מספר הפעמים המקס' שהיא יכולה להיקרא הוא , וכל פעם היא מקצה כמות מקום סופית (אובייקט ) סה"כ סיבוכיות מקום .
* מס' פעולות סופי כולל שמירת ערכים ושרשור לרשימה – סיבוכיות זמן .

פונקציה -

* מס' פעולות סופי כולל קידום  - סיבוכיות זמן 
* ריצה על רשימה של האיברים שצריך להדפיס + הפעלת מס' פעולות סופי כולל הדפסה, שחרור מרשימה, ושרשור לרשימה אחרת על כל אחד מהאיברים – סיבוכיות זמן , כאשר  זהו מס' המדינות המודפסות, וגודל הרשימה עליה אנו רצים.

בפונקציה –

* מס' פעולות סופי כולל, לגשת לאיבר, לשנות ערך, ניתוק מרשימה, ושרשור לרשימה חדשה – סיבוכיות זמן .

**שאלה 5 – מבני נתונים תחת סיבוכיות (20 נק')**

מעבדת המחקר על שם מר. ישלוכסף עובדת שעות ארוכות למציאת התרופה לקורונה. המעבדה מודעת לקיום זני קורונה שונים, ממוספרים לנוחיותנו לפי האינדקסים . שמם של הווירוסים נתון לכם במערך מחרוזות באורך . המעבדה פונה לעזרתכם על מנת למצוא מבנה נתונים יעיל לשמירת מנייה של מספר החולים בכל רגע נתון.

עליכם לממש מבנה הנתונים אשר תומך בפעולות הבאות:

מאתחלת את *המבני הנתונים הרלוונטיים.*סיבוכיות זמן: .

מקבלת כקלט את אינדקס הוירוס, ומוסיפה פציינט אחד לרשימת החולים בו.

סיבוכיות זמן: .

מחזירה את מספר החולים בוירוס ה-

סיבוכיות זמן: .

מדפיסה את שמם של הוירוסים המדבקים ביותר, בעלי מספר החולים הרב ביותר, לפי סדר יורד.

במקרה שישנם כמה פלטים אפשריים העומדים בתנאי הפונקציה, די בהדפסת

אחד מהם.

סיבוכיות זמן: .

מדפיסה את שמם של כל הוירוסים שעבורם מספר החולים הינו 0.

סיבוכיות זמן: , כאשר הוא מספר הוירוסים שיש להדפיס.

סיבוכיות המקום הנדרשת עבור המבנה הינה .

**הנחיות לפתרון:** עליכם תחילה לתאר מה כולל מבנה הנתונים שלכם **(רצוי להיעזר בציור).**

לאחר מכן, הסבירו את מימוש הפעולות, ונמקו עבור כל פעולה מדוע המימוש אשר הצעתם עומד בדרישות הסיבוכיות עבורה. אין צורך לכתוב פסאודו קוד.

**בהצלחה!**

**פתרון:**

**List**

**Values for sick people**

**Array**

**Sick people number**

elem

**Array**

**Viruses true order**

**Array**

**viruses**

**מבנה הנתונים יכיל:**

מערכים-

viruses (נתון)

List\_elm\* Sick people number מורכב משלשה מערכים (אלגוריתם אתחול הרצאה)

int Viruses true orderמורכב משלשה מערכים (אלגוריתם אתחול הרצאה)

רשימה ממוינת דו כיוונית שמכילה אברים מסוג elem - Values for sick people

**בelem** יהיו:

Set- מצביעים לאברים במערך Sick people number

ערך int

מצביעים אחורה קדימה ברשימה

מצביע לאיבר הגדול ברשימה- head ומצביע לאיבר הקטן- tail

**הסבר כללי:**

נדאג לכך שתמיד איברי המערך הראשונים ב- Viruses תמיד יהיו אלו שמספר החולים בהם שווה אפס ע"י שנחליף כל איבר שקידמנו אותו לראשונה ליהיות האחרון ברשימה לפי end\_zero

בנוסף נשמור את ההחלפה במערך Viruses true order כדי שנוכל לעקוב אחרי השמות

הדפסת כל הוירוסים עם אפס חולים היא מעבר על Viruses מ end\_zero עד תחילת המערך כי כך דאגנו שיהיו כל המאופסים בתחילת המערך.

ברשימה יהיו אברים עם ערך שכל איבר במערך Sick people number עם אותו ערך של חולים פשוט יצביע על האיבר ברשימה ולכן לפי שובך היונים באותו זמן יהיו עד n אברים ברשימה.

הרשימה תשאר ממויינת בO(1) כיוון שיש כל פעם תוספת של חולה אחד ז"א מוסיפים איבר למקום הבא ברשימה או שפשוט משתמשים בקיים.

הדפסת C הוירוסים הכי מדבקים היא פשוט מעבר מסוף הרשימה לC איברים.

אתחול :

מאתחלת שני מערכים (שהם בעצם 3\*2=6) בגודל n Sick people number  **,** Viruses true order כמו באלגוריתם של ההרצאה ולכן O(1).

בנוסף מאתחלת רשימה ריקה ז"א מחזיר שני פוינטרים head tail שמצביעים על null. O(1)

בנוסף end\_zero=n-1

**:**

**1)**

בודקת אם Viruses true order[i] מכיל ערך (ז"א נגשנו אליו כבר לפי האלגוריתם בהרצאה לאתחול מערך)

אם כן אז i= Viruses true order[i]

הפונקציה תבדוק אם לא השתמשנו בערך i] ] Sick people number (ז"א שהוא שווה null"") לפי האלגוריתם בהרצאה לאתחול מערך.

נעשה swap בין Viruses**[i],** Viruses[end\_zero].

בנוסף תעדכן את מערך Viruses true order[end\_zero]=i , Viruses true order[i]=end\_zero, .

**ו-**end\_zero--

בנוסף תבדוק אם קיים ערך 1 ברשימה לפי מצביע tail .

אם לא - תייצר איבר רשימה חדש שערכו 1 ומצביע דו כיווני בינו לבין i] ] Sick people number ונעביר את tail שיצביע עליו.

אחרת- נייצר אליו מצביע דו כיווני בינו לבין i] ] Sick people number

אחרת (קידמנו כבר את i] ] Sick people number בעבר)

ולכן נבדוק את המצביע i] ] Sick people number אם הערך באיבר הרשימה הבא אחריו שווה +1 הערך של האיבר הנוכחי

נבדוק אם הset של איבר הרשימה מכיל רק מצביע אחד (ז"א האיבר שלנו הוא היחיד שמצביע על איבר הרשימה הזה)

נמחוק את איבר הרשימה הזה ונעביר את המצביע הדו כיווני לאיבר הבא.

אחרת רק נעביר את המצביע לאיבר הבא.

אחרת נייצר איבר כזה חדש ברשימה ושוב נבדוק אם למחוק את האיבר הנוכחי או לא כנ"ל.

(בכל שלב נבדוק אם head מצביע על איבר ברשימה שפנינו אליו וייצרנו איבר אחריו אז נעביר את head שיצביע עליו כדי שהוא יצביע על המקסימלי, כמו כן לגבי tail לפני מחיקת איבר).

**:**

בודקת אם Viruses true order[i] מכיל ערך (ז"א נגשנו אליו כבר)

אם כן אז i= Viruses true order[i]

בודקת אם i] ] Sick people number מכיל ערך

אם כן מחזירה אותו,

אחרת תחזיר 0.

**:**

**תריץ לולאה c פעמים**

תבדוק אם head מצביע לערך חוקי

אם כן נדפיס את שמות c הוירוסים שהset של האיבר ברשימה הראשון

במידה ויש פחות נמשיך לאיבר הבא עם שארית מ-c עד שנגיע לnull

אם זה עדיין פחות מc נדפיס את כמות הוירוסים שנותרה מרשימת Virusesונתחיל מהוירוס מספר end\_zero

**:**

נדפיס את כל הוירוסים ברשימת Viruses החל מ Viruses [end\_zero] ומטה עד Viruses [0]