**

**

*נקודות חשובות*

* *חלק יבש*
  + *כשמוחקים אמן – אכן mlogn*
    - *למה – כי אנחנו רצים על המערך של השירים של אותו אמן, מוצאים לפי הסדר בעץ ה-AVL של האמנים ולמחוק את כל השירים שם – זה מסתכם ל-mlogn+m (צריך להסביר למה)*
  + *לוודא ולהסביר שאר ענייני סיבוכיות*
  + *לכתוב על טיפול בשגיאות ביבש*
  + *לציין את הפעולות בפירוט*
* *ווידוא ענייני C++*
  + *המרות*
    - *לוודא אם מותר מפורשות או שצריך STATIC*
  + *חריגות?*
    - *BAD\_ALLOC?*
    - *האם פספסנו משהו במימוש הנוכחי? (שלא מימשנו כ-exceptions)*
* *לטפל בקוד-*
  + *מימוש הרשימה*
  + *למחוק StreamList*
    - *חשבתי לממש באופן שמוחקים NODE-NODE – האם עומד בסיבוכיות?*
    - *צריך לעשות מתודה נוספת שמוצאת האם ה-key קיים?*
  + *יצירת StreamListNode*
    - *כרגע יוצר NODE רק אם יש לי עץ – לשנות?*
  + *Const ב-NODE – זה אומר שאי אפשר לשנות NODE, וצריך לאפשר את זה*
  + *להפוך פונקציות VOID לפונקציות ENUM?*
  + *לקבל דברים by value או by reference?*
  + *לשים לב לשימוש בבנאים והורסים – האם משתנה שהגדרתי יימחק אוט' בסיום הבלוק?*
    - *לראות שהשתמשנו נכון ב-new וב-delete*
  + *Include אחד לשני – artist-list*
    - *נכון להשתמש בהצהרות לפני שבכלל עשיתי include?*
    - *ענייני מערך של מצביעים בתוך מחלקה – עבור מערך השירים*
  + *לעבור ולסדר את ההערות והפונקציות – לוודא שהכל כמו שצריך*
    - *לראות אם יש פונקציות שאנחנו בכלל לא צריכים ופוגעות בנו*
  + *מה משמעות קובץ ה-main?*
  + *דברים שלא עובדים*
  + *טסטים*
    - *רשמיים*
    - *של אנשים*
    - *שרת*
  + *בדיקות אחרות*
    - *קונבנציות?*
    - *הערות?*

*פונקציות נדרשות-*

*Init*

*AddArtist*

*RemoveArtist*

*AddSongToCount*

*NumberOfStreams*

*GetRecommendedSongs*

*Quit*

***צריך לבדוק-***

* *אלגוריתמי חיפוש בעצי ה-AVL ועמידה בסיבוכיות*
* *מערך שירים – מכיל פוינטרים או ~~מחלקה מסוג SONG~~?*
* *עץ AVL של מספר השמעות – מכיל מספר או ~~מחלקה מסוג מספר השמעות~~?*

*מבנה הנתונים יורכב מהחלקים הבאים:*

***מחלקות כלליות***

1. *מחלקה גנרית של עץ AVL*
   1. *תומכת בכל הפעולות שצריך לעשות בעץ-*
      1. *הוספת צומת*
      2. *הוצאת צומת*
      3. *חיפוש צומת*
      4. *צריך לבנות נכון את האלגוריתמים?*
         1. *צריך לחשוב איך עושים את זה?*
   2. *כוללת גם NODE כללי גנרי*
      1. *שדות*
         1. *מצביע תבניתי ל-info (artist, song…)*
         2. *מצביע לבן ימני*
         3. *מצביע לבן שמאלי*
2. *מחלקה של אמן*
   1. *שדות:*
      1. *Int – Artist\_id? או שזה יהיה שם האמן?*
      2. *מערך – כלל השירים (מערך של מצביעים?)*
      3. *עץ AVL – מספר ההשמעות שיש לכלל השירים*
   2. *מתודות:*
      1. *Ctor*
         1. *בונה את השדה artist\_id דיפולטיבית*
         2. *בונה מערך בגודל N (מספר השירים בעת יצירת ה-Artist)*
            1. *הקצאה דינמית*
            2. *מכניס כפוינטר לרשימת ההשמעה 0*
         3. *בונה אובייקט מסוג AVL מסוג מספר השמעות?*
            1. *בהתחלה עם 0 צמתים (NULL)*
      2. *Dtor*
         1. *מוחק את השדה artist\_id דיפולטיבית*
         2. *מוחק את המערך*
            1. *מחיקה דינמית – delete*
            2. *צריך לדאוג למחיקה של הפוינטר שברשימה המקושרת המתאימה*
         3. *מוחק את האובייקט מסוג AVL*
            1. *מחיקה ע"י המתודה של AVL*
      3. *Cctor – למחוק (שלא תהיה אפשרות)*
      4. *Assign op – למחוק (שלא תהיה אפשרות)*
      5. *Getters*
         1. *Get\_song*
            1. *מחזיר SONG לפי אינדקס*
         2. *Get\_id*
            1. *מחזיר את ה-artistid*
      6. *Editors*
         1. *Add\_play\_to\_song*
            1. *מעלה ב-1 את מספר ההשמעות בשיר לפי אינדקס*
            2. *או*
            3. *מעביר את המצביע*
3. *מחלקה של שיר*
   1. *כנראה שמספיק להחזיק פוינטר – או שנעשה מחלקה (צריך לציין שדות ומתודות)*
4. *מחלקה של מספר השמעות?*
   1. *מספיר שה-key יהיה המספר, והאיבר עצמו יהיה עץ AVL של שירים (של INT המייצג SONGID)*

***מבנה הנתונים עצמו***

1. *X משתנים כלליים עבור מבנה הנתונים:*
   1. *שדכ*
2. *X – עץ AVL כללי של אמנים*
   1. *קיים אחד כזה באופן כללי*
   2. *עץ הבנוי מ-NODES מסוג artist*
3. *X – עצי AVL של מספר ההשמעות*
   1. *קיים אחד כזה עבור* ***כל artist***
   2. *עץ הבנוי מ-NODES מסוג song*
4. *X – עצי AVL של שירים*
   1. *קיים אחד כזה עבור* ***כל מספר השמעה***
   2. *כל צומת – SONGID*
5. *X – רשימה מקושרת של "מספר השמעות"*
   1. *נוצרת בעת האתחול הראשוני init*
   2. *קיימת אחת כזו*
   3. *מכילה איברים מסוג עץ AVL מסוג Artist*
      1. *(שיש להם את אותו מספר השמעות לפי האיבר)*
6. *דגע*

***אלון חושב על דברים:***

*דברים שאני רוצה לחשוב עליהם-*

* *העץ – רץ על האלמנטים בצמתים*
* *לחשוב איפה מימשתי כאילו אני משתמש ב-NODE של עץ ה-AVL עצמו ולא באלמנט*
* *צריך גם GetLast?*
* *למה אנשים משתמשים ב- Unique\_pointers? Shared\_pointers?*
* *מתי להשתמש בnested classes*

*דברים שאולי היה נכון יותר לעשות-*

* *צריך לעשות שיהיה גם int עבור כל צומת של "מספר השמעות"? ואז לעשות מחלקה כזו?*
* *הוצאנו את את NODE מה-PRIVATE*
  + *מחיקה ידנית של NODES בSTREAMLIST?*
* *לשנות מימוש של הרשימה לאיטרטור?*

*איטרטור-*

* + *דירפרנס ל-\**
  + *האם חרג*
  + *העמסת אופרטורים*

**GetRecommendedSongs:**

נראה שהפונקציה אכן עונה על התנאי של הסיבוכיות שהוא O(m). אנחנו מתחילים בצומת האחרון ברשימת ההשמעות שלנו (מכיוון שיש לנו פוינטר אליו ההגעה לצומת היא O(1))שהיא מייצגת את מספר ההשמעות המקסימלי שיש לנו במערכת (נסמנו M) בצומת זו יש לנו עץ של פוינטרים לעצים המתאימים לM שירים לאומנים אשר יש להם לפחות שיר אחד עם M השמעות. אנחנו ניגשים לאיבר הראשון בעץ הפוינטרים (מכיוון שיש לנו פוינטר first תחת ה-AvlTree זה נעשה ב-O(1))- זה בעצם פויטר לעץ המתאים של האומן עם האינדקס הכי קטן שיש לו M השמעות (נסמנו ) . בעץ של השירים עם M השמעות תחת נעבור את השירים לפי הסדר (שוב מתחילים מהשיר עם האינדקס הכי קטן,הגעה אליו זה O(1) בגלל first ), כאשר על כל שיר שנעבור נשים את הערכים המתאימים למערכים, מכייון שיש לנו count שמתחיל מ-0 ושאותו אנחנו מגדילים ב-1 לאחר כל מעבר על שיר נדע תמיד את האינדקס המדוייק שבו אנחנו צריכים לשים את הערכים במערך. אנחנו נעבור על כל העץ (במקרה וnumSongs גדול ממספר השירים שיש ל- עם M השמעות ) או שנעבור רק על חלק מהעץ. נסמן את מספר הצמתים בעץ שנעבור עליהם ב-. מטענת העזר סך הפעולות שנעשה הוא O(). וזה נכון לכל העצים שלעיהם נעבור. מכיוון שאנחנו רצים בלולאות while עם count שמגדילים אותו ב-1 לאחר כל מעבר ומכיוון שמעבר מצומת לצומת הקודמת ברשימת השמעות היא O(1) על שיר נעשה סך הכל מספר פעולות של:

.

כנדרש, כאשר השוויון הראשון נובע מטענה שעשינו בתרגול.

טענת עזר: נראה שבמידה ואתה מתחיל בצומת בעל האינדקס הכי קטן בעץ, ואתה עובר(מדפיס) לפי סדר r צמתים, אז תעשה סך הכל O(r) פעולות.

הוכחה: מכיוון שאנחנו מתחילים בצומת המינימאלי, הצומת הזו נמצאת בתחתית העץ (העלה הכי שמאלי) וכדי להגיע לצומת ה-r נעשה (מספר העליות שביצענו בעת המעברים)O, עתה מכיוון שאנחנו עוברים לפי סדר בכל עלייה כזו נצטרך או שלהדפיס את הצומת (במידה ועלית מבן שמאלי) או שלא (עלייה מבן ימני) ואז כבר הדפסת את הצומת הזו (כי הבן הימני יותר גדול מהצומת הנוכחי ואנחנו עוברים לפי סדר). לכן מספר העליות שסך הכל ביצענו הוא O(r), ולכן סך הפעולות שנעשה הוא O(r). כפי שרצינו להראות.

***Quit:***

*ב-quit מה שאנחנו עושים בעצם זה לקרוא לדסטרטור של ה-MusicManager. שהוא קורא לדיסטרקטור של של העץ אמנים שלנו ולדיסטרקטור של הרשימת השמעות. נראה עבור כל אחד שהוא עומד בתנאי הסיבוכיות.*

*עץ אמנים:*

*בעץ אמנים יש לנו עץ ובו n צמתים (כמספר האמנים), נעשה זום אין לצומת כזו, הצומת הזו מייצגת אומן מסויים (), באומן הזו יש עץ השמעות(num\_of\_streams\_tree) כאשר ה-key של כל צומת זה מספר ההשמעות וה-element של כל צומת זה עץ עם השירים שיש לאומן המסויים עם מספר השמעות של key.*

*נצטרך למחוק את העץ שירים הזה, המחיקה של עץ כזה היא O( כאשר זה מספר השירים שיש לאמן עם השמעות, בנוסף בסופו של דבר נצטרך למחוק את המערך full\_songs\_list ומכייון שזה בסופו של דבר פונינר מחיקתו זה ב- O(1) (מחקית האיברים שהוא מצביע אליהם נעשית דרך המחיקה של הרשימה המקושרת שלנו). לכן סך הכל עבור מחיקה של אמן מעץ האמנים שלנו נעשה מס' פעולות שהוא*

*כאשר זה מספר השירים שיש לאמן a. זה עבור אמן ספציפי, עבור מחיקה של כל עץ האמנים נעשה באותה דרך:*

*.*

*רשימת השמעות:*

*נצטרך לעבור צומת צומת ובכל צומת למחוק את הצמתים בעץ הפוינטרים המתאים למספר ההשמעות של הצומת בו אנחנו נמצאים (את הדברים אליהם אנחנו מצביעים אנחנו מוחקים בעת מחיקת עץ ההשמעות שלנו). בכל צומת ברשימת ההשמעות נעשה מטענת העזר (מספר הצמתים בעץ הפונטרים) O ולכן סך הכל נעשה (סך הפויטרים)O וזה לכל היותר O(n+m)*

*שאלה 4*

*להלן ציור לדוגמה הממחיש את מבנה הנתונים שלנו:*

*Countries*

***Index***

***d***

***ptr***

*DaysPast N D Country*

*DaysToElections*

*List*

*\* ניתן לראות ציורי עזר עבור כל אחת מהפעולות בסוף התיאור של כל פעולה.*

*מבנה הנתונים יורכב מהחלקים הבאים:*

1. *שלושה משתנים כלליים עבור מבנה הנתונים:*
   1. *N – מקבל את הפרמטר N עבור מספר המדינות המקסימלי*
   2. *D – מקבל את הפרמטר D (הוא ) עבור מספר הימים המקסימלי בין מערכות בחירות*
   3. *DaysPast – מתאר את מספר הימים שעברו מיום אתחול המבנה (כלומר מיצירתו).*
2. *Countries – מערך באורך N (אשר נתון לנו בעת אתחול המבנה) של איברים מסוג Country (הטיפוס יתואר בהמשך), שיחזיק את הקלט של המשתמש עבור כל מדינה*

*\* מערך זה יתחיל באינדקס 1, לשם נוחות.*

1. *Country – struct המייצג כל מדינה במערך Countries אשר המשתמש מכניס כקלט.*

*שדות ה-struct:*

* 1. *Index – האינדקס של המדינה*

*(זו למעשה הדרך שלנו לזהות מדינה בשאלה – ע"פ אינדקס; אם היינו מעוניינים, היינו יכולים ליצור מילון key-value המקשר בין אינדקס (key) לשם המדינה (value) ובכך לקבל את שם המדינה בשלמותו, אך אין דרישה כזו בשאלה – ועל כן מעתה ולאה בשאלה מדינה תזוהה ע"פ האינדקס שלה)*

* 1. *– המחזור בין בחירות כלשהן לבחירות הבאות*
  2. *d – מספר הימים שנותרו עד לבחירות הבאות במדינה (מיום הכנסת המדינה למבנה)*

*זהו מצביע לאיבר המתאים (מסוג רשימה מקושרת) במערך DaysToElections. יתואר במפורט בהמשך*

* 1. *ptr – מצביע לאיבר המתאים (מסוג integer) בתוך כל רשימה מקושרת שבמערך DaysToElections, יתואר במפורט בהמשך*

1. *DaysToElections – מערך באורך D (אשר נתון לנו בעת אתחול המבנה) של איברים מטיפוס רשימה מקושרת דו-כיוונית מעגלית (רשימה מקושרת מהסוג שלמדנו).*

*הרשימה המקושרת תהיה למעשה רשימה של integers, אשר מייצגים את ה-i של כל מדינה, כאשר המיקום של הרשימה במערך (האינדקס של האיבר במערך) יציין את מספר הימים עד לבחירות הקרובות של המדינות שברשימה. יתואר במפורט בהמשך*

*נזכיר כמה נקודות רלוונטיות להיגיון המסדר מאחורי מבנה הנתונים:*

* *DaysPast – בכל יום שעובר, נגדיל את המשתנה ב-1 (DaysPast++) על מנת לציין זאת.*
* *במימוש הנ"ל לא ייתכן כי איבר כלשהו מטיפוס Country יצביע ליותר מאיבר אחד שתחת איזו שהיא רשימה מקושרת שבמערך DaysToElections (ניגע בדבר בהמשך)*
* *נשים לב כי כאשר נתייחס בהמשך למשתנה DaysPast, פעמים רבות נסתכל עליו ב-*

*נתאר את אופן הביצוע של כל אחת מהפעולות הנדרשות בשאלה:*

***Init(N, D)***

*על מנת לבצע את פעולת האתחול של המבנה בסיבוכיות נאתחל את כל אחד מהמערכים הנתונים (Countries ו-DaysToElections) בצורה שנלמדה בהרצאה, ע"י שימוש ב-2 מערכים נוספים (מערך עזר ומחסנית) לכל מערך.*

* *Countries – יאותחל כמערך של N איברים לא מאותחלים ובהם ערך זבל (כאשר בהמשך, בעת הכנסת מדינה, ניצור עבורה איבר מסוג Country ונסמן כי האינדקס שלה מאותחל).*
* *DaysToElections – באופן דומה, יאותחל כמערך של D איברים לא מאותחלים ובהם ערך זבל (כאשר בהמשך, כשנכניס מדינה עם d שאינו קיים עדיין – ניצור עבורו רשימה מקושרת, ונסמן כי האינדקס שלה מאותחל).*

*בעת כל אתחול, המשתנים D ו-N יאותחלו לפי הפרמטרים שנקבל, וכן המשתנה DaysPast יאותחל ל-0 (שכן זהו היום בו אותחל המבנה, ולכן גם היום בו התחילה הספירה).*

*עמידה בסיבוכיות :*

*כפי שציינו, הפעולה עומדת בסיבוכיות הנ"ל כתוצאה משימוש בשיטת האתחול שנלמדה בהרצאה (בתוספת כמה פעולות אתחול משתנים שמספרן קבוע, שלא יפגעו בסיבוכיות ).*

*\* ציור לדוגמה*

*Init(16, 16)*

*Countries*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

*DaysPast N D*

16 16 0

*DaysToElections*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

***AddCountry(i, , d)***

*כך תיראה הוספת המדינה i למבנה הנתונים:*

*ניצור איבר חדש מטיפוס Country אשר יכיל בשדותיו-*

* *index – המספר i שקיבלנו מהמשתמש*
* *– הפרמטר שקיבלנו מהמשתמש*
* *d – לפי d שהתקבל כפרמטר מהמשתמש, נגדיר מצביע לאינדקס שמספרו (DaysPast+d)%D במערך DaysToElections.*
* *ptr:*

*נתבונן בשני מקרים:*

* + *בהנחה שלא קיים איבר (מטיפוס רשימה מקושרת) באינדקס (DaysPast+d)%D במערך DaysToElections:*

*ניצור רשימה מקושרת שמקומה יהיה באינדקס הנ"ל, והאיבר הראשון בה יהיה i – כאשר הוא אינו מצביע לאיבר עוקב או קודם שכן הוא היחיד ברשימה זו כרגע.*

* + *בהנחה שקיים איבר (מטיפוס רשימה מקושרת) באינדקס (DaysPast+d)%D במערך DaysToElections:*

*ניצור איבר חדש, i, ונקשר אותו (דו-כיוונית עם מצביעים) לאיבר האחרון ברשימה המקושרת הנ"ל. (כך למעשה נוצרת לנו רשימה של integers המייצגים אינדקסים של מדינות, אשר מועד הבחירות הקרובות שלהם הוא בעוד d ימים)*

*עמידה בסיבוכיות :*

*נשים לב כי אנו מבצעים מספר סופי וקבוע של פעולות-*

* *הכנסת הפרמטרים הרלוונטיים*
* *יצירת מספר סופי מקסימלי של איברים מטיפוסים מוכרים (מצביע, Country, רשימה מקושרת, ואיבר ברשימה הזו)*

*אנו לא תלויים במקרה זה בגודל הקלט, אלא רק במימוש הפנימי של יצירת האיברים הנ"ל – שיתבצע בסופו של דבר תוך מספר סופי של פעולות.*

*\* ציור לדוגמה*

*DaysPast = 1*

*AddCountry(13, 10, 1)*

*Countries*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

***13***

***10***

*מצביע*

*מצביע*

*DaysPast N D Country*

16 16 1

*DaysToElections*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

*13*

*List*

***PredateElections(i, d)***

*עלינו להקדים ב-d את מספר הימים עד לבחירות הקרובות במדינה ה-i – לכן נתאים את מימוש הפעולה למבנה הנתונים שבחרנו:*

*נתבונן במדינה ה-i. בשדה d שלה ישנו מצביע לאיבר מטיפוס רשימה מקושרת אשר במערך DaysToElections. האינדקס של האיבר בתוך המערך מסמן כמה ימים נשארו כרגע עד לבחירות הקרובות במדינה, ולכן אם נחסיר (מודולו D) מאינדקס זה את d שקיבלנו כפרמטר (נאמר בפיאצה כי ניתן להניח שחיסור זה לא ייתן מספר שלילי) – נוכל לקבל את מספר הימים החדש שנשארו עד לבחירות הקרובות (נסמנו לעת עתה tempd).*

*כעת, על מנת לממש את הפעולה, נשנה את המצביע שבשדה ה-d כך שבמקום להצביע לאיבר באינדקס שאליו הוא הצביע, יצביע כעת לאיבר באינדקס tempd במערך DaysToElections.*

*בנוסף, בשדה ptr של המדינה ה-i ישנו מצביע לאיבר בתוך הרשימה המקושרת המדוברת. איבר זה שנסמנו x (שמכיל למעשה האינדקס של המדינה i כ-integer) מצביע על איבר אחר ("מדינה אחרת"), וכן עליו מצביע איבר אחר ברשימה (שוב, "מדינה אחרת"). ייתכן מצב גם בו x אינו מצביע על איבר אחר או שלא מצביעים עליו – במקרה הזה המצביעים הם NULL וזה בסדר מבחינתנו.*

*עתה, נחליף את המצביעים הנ"ל למצביעים מתאימים ברשימה המקושרת שבאינדקס tempd – האיבר האחרון שבה יצביע על האיבר x, ואילו x לא יצביע על איבר נוסף שכן הוא האחרון ברשימה כעת.*

*על מנת להשלים את מימוש הפעולה – נוודא כי המצביעים של האיברים הקודם והעוקב ברשימה הראשונה יצביעו זה על זה. אם אין איבר קודם/עוקב – המצביעים הם NULL כפי שציינו. אם היה זה האיבר היחיד ברשימה – נוכל לדאוג לכך שנסמן את הרשימה כריקה (נניח, ע"י איבר דמה ריק).*

*עמידה בסיבוכיות :*

*נשים לב כי גם כאן, אנו מבצעים מספר סופי וקבוע של פעולות-*

* *חישוב פעולת החיסור בין d הפרמטר ל-d השדה של המדינה*
* *החלפת מספר סופי של מצביעים (המצביעים לרשימה המקושרת, והמצביעים לאיברים בתוך הרשימה המקושרת)*

*אנו לא תלויים במקרה זה בגודל הקלט, אלא רק במימוש הפנימי של חישוב פעולת החיסור ושל החלפת המצביעים שציינו – שיתבצעו בסופו של דבר תוך מספר סופי של פעולות.*

*\* ציור לדוגמה*

*DaysPast = 1*

*PredateElections(13, 1)*

*Countries*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

***13***

***10***

*מצביע*

*מצביע*

*DaysPast N D Country*

16 16 1

*DaysToElections*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל ריק

*13*

*List*

***ElectionsToday()***

*נרצה להדפיס את האינדקסים של כלל המדינות שבהן נערכות הבחירות ביום הפעלת הפונקציה (כאשר מספר המדינות מסומן ב-k), כאשר נשים לב כי הפונקציה רצה בכל יום.*

*נשים לב כי כאשר אנו רוצים להשתמש בפעולה עבור יום מסוים, אנו מחפשים את המדינות עבורן יש 0 ימים עד לבחירות הבאות. דרך אפשרית למצוא אותן הייתה לחסר בכל יום 1 מ-d עבור כל המדינות במבנה – אך הדבר לא היה עומד בדרישות השאלה (שכן אם יש N מדינות – היינו נאלצים לעשות סדר גודל של פעולה עבור כל מדינה, כלומר סדר גודל של N פעולות, כלומר ).*

*לשמחתנו, ישנה דרך שקולה שעוזרת לנו לבצע פחות פעולות:*

*ציינו קודם לכן כי בכל יום, המשתנה DaysPast גדל ב-1 – כלומר מתחיל מ-0 ביום אתחול מבנה הנתונים, ולאחר שבוע למשל יגיע ערכו ל-7. מכאן, שמדינה עבורה d=7 תגיע בעוד שבוע בדיוק ליום בו הבחירות בה יתקיימו, ונרצה להדפיסה.*

*בעצם, הדבר אומר כי אנו מחפשים להדפיס את המדינות בהן d מצביע על האינדקס שווה ל-(DaysPast)%D ביום הפעלת הפונקציה.*

*)נסביר זאת ע"י דוגמה – נניח כי אנו נמצאים ביום ה-a, כלומר DaysPast=a. אם נכניס מדינה כלשהי למבנה הנתונים עם הפרמטר d=3, אזי השדה d תחת ה-Country יצביע לאינדקס (a+3)%D (לפי הגדרת הפעולה AddCountry). לאחר 3 ימים נקבל כי DaysPast=a+3 – כלומר נקבל כי האינדקס שווה ל-DaysPast).*

*כלומר, אחרי ההסבר הארוך, מצאנו דרך בעזרתה, בהינתן המשתנה DaysPast, נוכל להגיע לרשימה המקושרת המכילה איברים המייצגים את כל המדינות שהבחירות בהן מתקיימות היום.*

*דרך זו דורשת מאיתנו חישוב מינורי שלוקח מספר פעולות קבוע, ולכן היא .*

*\* אם באינדקס שקיבלנו מדרך זו אין רשימה מקושרת (כלומר הערך הוא ערך זבל), סימן שאין מדינות שמתקיימות בהן בחירות היום.*

*כעת, כשאנחנו יודעים מהי הרשימה המקושרת המבוקשת, נוכל לרוץ עליה (מהאיבר הראשון ועד לאיבר האחרון שלה), כאשר עבור כל איבר נדפיס את המידע שבו – שהוא למעשה האינדקס של המדינה שרצינו להדפיס.*

*לאחר מכן, עלינו לשים לב כי עבור כל המדינות הנ"ל השדה d השתנה – התקיימו היום בחירות, ולכן הבחירות הבאות יהיו בעוד ימים. לכן, עבור כל איבר עליו רצנו נרצה לא רק להדפיס את i, אלא גם למחוק את המצביעים שלו לאיבר הקודם/הבא ומהאיבר הקודם/הבא ובמקומם לשים מצביעים דומים עבור האיברים שברשימה שבאינדקס (DaysPast+)%D במערך DaysToElections, וכן לשנות את המצביע שבשדה d של Country בהתאם (כלומר – לעשות פעולות דומות מאוד לאלו שעשינו עבור AddCountry).*

*לסיום - נוכל לדאוג לכך שנסמן את הרשימה כריקה.*

*עמידה בסיבוכיות :*

*נתבונן בסך הפעולות שנבצע עבור הפעולה:*

* *ציינו קודם, בהסבר על מימוש הפעולה, כי נעשה פעולות על מנת למצוא את האינדקס של הרשימה המקושרת המבוקשת.*
* *נשים לב כי ברשימה המקושרת הנ"ל, יש בדיוק k איברים – הרי שכל איבר מייצג בדיוק מדינה אחת מהמדינות שנרצה להדפיס, וכל המדינות הנ"ל מיוצגות על ידי איבר כלשהו ברשימה הזו. אזי, כאשר נרוץ על איברי הרשימה המקושרת, למעשה נרוץ על k איברים.*
  + *עבור כל איבר שכזה, נבצע-*
    - *הדפסת המידע שבתוך האיבר – האינדקס של המדינה*
    - *החלפת מספר סופי של מצביעים (המצביעים לרשימה המקושרת, והמצביעים לאיברים בתוך הרשימה המקושרת)*

*כלומר, עבור כל איבר ברשימה המקושרת נבצע מספר סופי של פעולות.*

*אזי – עבור הריצה על כל הרשימה המקושרת נבצע k פעולות כפול מספר סופי של פעולות - .*

*מכאן, שסיבוכיות הזמן הכללית שלנו תהיה:*

*\* ציור לדוגמה*

*DaysPast = 1*

*ElectionsToday()*

*Countries*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל

***13***

***10***

*מצביע*

*מצביע*

*DaysPast N D Country*

16 16 1

*DaysToElections*

זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל זבל ריקריק

*13*

*List*

*Printed: “13”*

*לבסוף, נוכל לראות כי סיבוכיות המקום שלנו אכן עומדת ב-:*

*המערך Countries יכיל לכל היותר N איברים מטיפוס Country. כיוון שהטיפוס הנ"ל מכיל בתוכו שדות בגודל 2 מצביעים ו-2 משתנים, המערך יתפוס לכל היותר זיכרון בגודל 4N, כלומר .*

*המערך DaysToElections יכיל לכל היותר D איברים מטיפוס רשימה מקושרת דו-כיוונית. כל רשימה כזו תכיל איברים שמורכבים מ-int ומשני מצביעים לכל היותר, וכן מספר האיברים הכולל בכל הרשימות בכללי יהיה לכל היותר N (שכן האיברים ברשימות המקושרות מייצגים מדינות, ולא יהיו איברים המייצגים את אותה מדינה). נקבל מכאן כי המערך יתפוס לכל היותר זיכרון ב-.*

*בנוסף, נשמור עוד 3 משתנים – N, D, DaysPast. כלומר – נצטרך עוד זיכרון בגודל קבוע, אזי .*

*לכן, חישוב סיבוכיות המקום הכולל יהיה:*