**מנמ"א - ממן 14**  
204122188  
ראשית נסמן:  
 - "מבנה הנתונים" אליו שייכים איברי הקבוצה הנתונה. .  
 - קבוצת פונקציות הגיבוב המדוברות..  
 פלט הפונקציות הוא מספר שלם בין ל .  
 – טבלת הביטים הנתונה. .

1. הסיכוי הוא 0. נסביר:  
   היות ועבור כל איבר בקבוצה מסומנים ב1 כל המקומות המותאמים לו בT ע"י פונקציות ,  
   לא יתכן כי בעת חיפוש ימצא באחד המקומות האלו 0 (זאת בהנחה שאין פונקציית מחיקה),  
   ולכן הבדיקה שלו תחזיר תמיד אמת.
2. לפי ההדרכה ראשית נחשב  
   מה הסיכוי שעבור איבר כלשהו (שלא בהכרח ב ),  
   קיימת פונקציה מסויימת, , שתחזיר עבורו מספר, (שהוא אינדקס מסויים ב),  
   כך שלכל וכל מתקיים .  
     
   כלומר- היות ול יש סיכוי של להתקבל ע"י כל אחד מפעולות   
   נחשב מה הסיכוי שכל אחת מפעולות על כל אחד מאיברי נתנה תא שאיננו :  
     
   הסיכוי שתא זה לא נבחר עבור ריצה מסויימת של הוא .  
   בדיקה זו צריכה להחזיר **לכל** הריצות של כל הפונקציות על מסיים איבר שונה- ולכן:  
   לכל איבר יש סיכוי של לא להיות מנותב ע"י אף פונקציה אל .  
     
   היות ותנאי זה נדרש עבור כל , נדרש שנחזור על התוצאה הנ"ל פעמים,  
     
   כך שבסה"כ הסיכוי שתא : .   
     
   כעת, הסיכוי שאיבר שאיננו במבנה הנתונים ()  
   יקיים לכל .  
   אז לפי טענת העזר הסיכוי שעבור מסויים- לא יתן תא שיש בו הוא היות והחיפוש דורש שעל מנת שאיבר יוכרז כאיבר ב בכל החיפושים יתקבל עלינו לחזור על תנאי זה עבור כל הפונקציות  
     
   לכן בסה"כ הסיכוי הוא . משל.
3. לפי סעיף ב' התשובה היא:
4. תיעוד הmain:  
   - מבני הנתונים העיקריים הם:  
    TABLE: טבלת הגיבוב (המכילה 0ים או 1ים)  
    db\_list: רשימת המחרוזות שיגובבו אל תוך הטבלה.  
    checks\_list: רשימת המחרוזות שנבדוק אם קיימים בה איברים  
    שכל גיבוביהם מסומנים בטבלה.  
   - הגישה הכללית לפתרון:  
    הפיכת מאגרי הנתונים לרשימות של מחרוזות,  
    גיבוב אחת מהן אל הטבלה, בדיקת השניה אל מול הטבלה כמתואר בתיאור checks\_list.  
   - תיאורי פונקציות וקשרים בין הפונקציות:  
    מפורטים בתוכנית.  
   - זמני ריצה של הפונקציות:  
    כל הזמנים מחושבים לפי ההנחה ש ו- .

* :  
  אנו יוצאים מנקודה הנחה שפונקציית הגיבוב היא .
* :
* :  
  לפי החישוב של הזמן הוא
* :

-הבהרה לגבי השימוש ב % (מודולו) בפונקציה inserts:  
כמות המספרים שhash יכול לייצר היא   
גדולה מכמות המספרים שטבלת הגיבוב יכולה לקלוט - .   
לכן- על מנת שלא נקבל ערכי hash הנמצאים מחוץ לטבלה, אך בכל זאת נשמור על פיזור אחיד ככל האפשר- נעזר בפונקציית מודולו על כל פלט של פונקציית hash.

**דוגמא להרצת הmain:**

דוגמת הרצה אחת של main בנויה אל תוך main. פשוט צריך להריץ אותה כפי שהתקבלה.  
(לעוד דוגמות יש להעביר מתיקיית "More checks" את קבצי ה-.txt הרצויים אל התיקיה הראשית ולשנות את שמות הקבצים).