שיר אלבז, 204405690
גל ארוס, 204372619
מתן דנינו,304802887

עבודה 2 דחיסת נתונים : ArithEncoderDecoder

COMPRESS:

בתחילת הפונקציה ניתחנו את ההסתברויות של כל תו במחרוזת ואחסנו את השכיחויות במערך מסוג DOUBLE ששמו- fixedFreq,ואת הערכים של התווים אחסנו במערך נוסף מסוג CHAR בשם- dictionary כך שהערך באינדקס i ב-fixedFreq מהווה את ההסתברות בטקסט של התו באינדקס i ב-dictionary.

הערה : השתמשנו בדיוק של שלוש ספרות אחרי הנקודה על ידי פונקציית round

לאחר ניתוח התדירויות התחלנו לבצע את פעולת הדחיסה באמצעות שיטת ה-scaling ואת ערכי המשתנים אחסנו במשתנה מסוג BigDecimal המיועד לאחסון מספרים דצימליים גדולים על מנת לא לאבד מידע.

את מתודת הדחיסה מימשנו באופן רקורסיבי כך שבכל קריאה נוצר גרף SCALING באמצעותו מקודד תו אחד בכל פעם- RECDEC באופן הבא :

1. יצירת גרף SCALING על ידי מתודה makeGraph
2. מציאת התו הבא שעלינו לקודד על ידי מתודה currentCharIx
3. ביצוע קריאה רקורסיבית עד להגעת סוף המחרוזת- ז"א כי קידדנו הכל
4. במידה והגענו לסיום המחרוזת נחזיר את הקידוד על פי הנוסחה : (high+low)/2

לאחר שדחסנו את המחרוזת לקוד דצימלי המתבסס על קידוד אריתמטי, ביצענו המרה לבינארי ואחסנו בתוך משתנה מסוג STRING.

לאחר ההמרה שרשרנו למחרוזת המייצגת את הקידוד הבינארי את המילון וההסתברויות.

DECOMPRESS:

תחילה חילצנו את המילון וההסתברויות מתוך הקידוד הבינארי לתוך המערכים fixedFreq ו dictionary בהתאמה.

בשלב הבא, המרנו את המספר הבינארי למספר דצימלי באמצעות המתודה toDecimalCode המחזירה את הקידוד האריתמטי המקורי .

את פעולת החילוץ ביצענו באופן רקורסיבי באמצעות המתודה DecRec באופן הבא :

1. יצירת גרף SCALING על ידי מתודה makeGraph
2. חישוב הטווח מתוך הגרף על ידי מתודה range
3. עדכון ערכי ה-high,low המהווים את גבולות גרף הscaling
4. חילוץ תו מתוך הקידוד לתוך מחרוזת המייצגת את הפענוח בשם- uncoded
5. ביצוע קריאה רקורסיבית עם המערכים המעודכנים כל עוד לא הגענו לגודל המחרוזת המקורית.

בסיום פעולת החילוץ מאחסנים את המחרוזת המפוענחת לתוך מערך בשם- output_names[i]