קורס מבני נתונים

תרגיל תכנות מס' 2

קבוצה 15

מגישים:

תומר מוצרי 206782641

זוהר עציץ 318590791

פלג עז-ארי 314632019

שאלה 1: מימוש תור:

מימוש קוד כולל התיעוד נמצא בקובץ q1_15

חישוב סיבוכיות אלגוריתם לזיהוי התקפה:

סיבוכיות מיון רשימה A לפי הנתון הוא O(nlogn) .

לולאת ה While - הראשונה רצה על כל n האיברים.

לולאת ה While - הפנימית רצה עד כמות ההודעות ברצף שחשודות בהתקפה, נקרא לכמות הזאת k. כל הפעולות בתוך כל אחת מהלולאות עולות O(1) לכן הלולאה הפנימית תרוץ k פעמים כפול n של הלולאה החיצונית לכן – הסיבוכיות O(n*k) .

בנוסף Dequeue עולה O(1) .

לסיכום, נקבל O(nlogn + n*k) .

במידה וכל האיברים הנתונים הם חלק מההתקפה אז k=n ולכן הסיבוכיות תהיה O(n^2) .

שאלה 2: מימוש ערימה:

1. מימוש קוד כולל תיעוד נמצא בקובץq2_15
2.
    1. מימוש קוד כולל תיעוד נמצא בקובץ q2_15
    2. מימוש קוד כולל תיעוד נמצא בקובץ q2_15, הערך האופטימלי שהתקבל הוא 749302
    3. תחילה פיצלנו את הבעיה לשלושה מקרים:
        a. אם הרשימה המתקבלת היא ללא חברות אזי אין חברות לאחד לכן שווי המס שנצטרך לשלם הוא 0.
        b. אם הרשימה המתקבלת כוללת חברה אחת, אז לא יהיה איחוד ולכן לא נצטרך לשלם מס, ושווי המס יהיה 0
        c. אם ברשימה המתקבלת 2 חברות או יותר, מתבצעות הפעולות הבאות:
            i. יוצרים ערימת מינימום בהסתמך על הרשימה שהתקבלה

ii. מוצאים את שתי החברות בעלות הערך המינימלי באמצעות ה-delete_min()
iii. מחברים את ערך החברות על מנת לחשב את שווי המס שנצטרך לשלם על האיחוד, למשתנה הסוכם את סה"כ המס שנחשב
iv. מחזירים את שווי החברה המאוחדת לערימה באמצעות insert()
v. פעולות 2 – 4 חוזרות על עצמן עד שנגיע לערימה בגודל 1 (כלומר עד לשווי החברה המאוחדת הכוללת)

נסביר מדוע בחרנו במימוש אלגוריתם זה:

המטרה באלגוריתם הוא תשלום מס מינימלי עבור איחוד כלל החברות. לכן, מכיוון שהתשלום על איחוד 2 חברות הוא מיידי, ובנוסף תשלום עתידי על כל איחוד שלהם עם חברה אחרת – בסיס האלגוריתם נבנה על ההנחה כי יש לאחד את החברות בעלות הערך המינימלי קודם. ובכך להבטיח כי המס שנשלם יהיה האופטימלי מבין כל אופציות סדר האיחוד האפשריות.

4. סיבוכיות האלגוריתם:
   a. בניית הערימה בסיבוכיות של O(n)
   b. מחיקת איבר באמצעות delete_min() פעמיים בסיבוכיות O(2n *logn)
   c. הכנסת החברה המאוחדת חזרה לערימה בסיבוכיות O(n*logn) פעמים
   d. סה"כ סיבוכיות האלגוריתם היא O(n*logn)
5. קוד האפמן הוא שיטה לקידוד סממנים ללא איבוד נתונים.

הקוד מספק דחיסת נתונים מרבית, כלומר מאחסן את הסימנים במספר מזערי של סיביות על פי קריטריון מיוחד. השיטה מתבססת על הקצאת אורך משתנה לסימנים על פי שכיחותם, כך שסימן נפוץ יוצג באמצעות מספר קטן של סיביות.

האלגוריתם עובד באופן הבא:

א. ראשית יש למצוא מה התדירות עבור כל אות במחרוזת. האלגוריתם בונה מעין ערימת מינימום כך שברמה התחתונה ביותר העלים מחזיקים את האות ובנוסף את הערך המתאים.
ב. בונים את הערימה עם האותיות בתדירות הנמוכה ביותר, האבות מחזיקים את סכום הערכים של הבנים שלהם, וממשיכים "לדחוף" לערימה את האות הבאה עם התדירות הנמוכה מבין האותיות שנשארו. בעבור אותיות בעלי תדירות גבוהה מהאב שחושב כבר קודם, "נפתח ענף חדש" שהאות תיכנס תחתיו והאב החדש יסכום את האב הקודם יחד עם תדירות האות החדשה.
ג. לאחר שהמחרוזת מקודדת ניתן לקרוא אותה שוב ע"י קריאת הקידוד עפ"י העץ המתאים.
ד. כך ש: 1 מייצג פנייה ימינה אל הבן הימני ו-0 מייצג פנייה שמאלה אל הבן השמאלי.

משתמשים ב- Huffman coding בעיקר באלגוריתמי קידוד של קבצי , JPEG, ZIP PKZIP, GZIP ו MP 3-כדי לדחוס את הקובץ. בנוסף בהעברת נתונים כמו פקס והודעות כדי להוריד את זמן שליחת הנתונים.

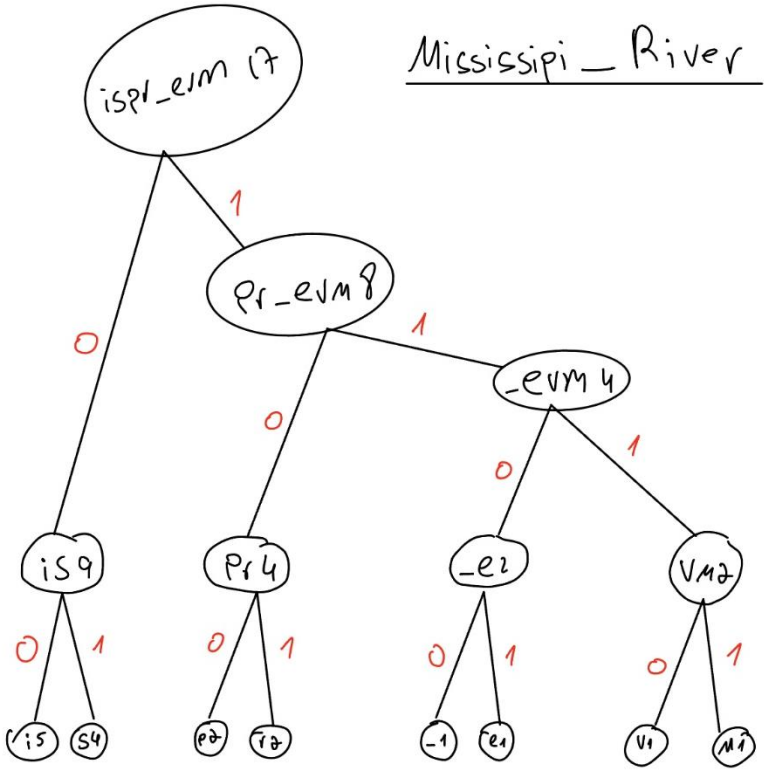
דוגמא :

נכתוב את המילה Mississippi River בצורת Huffman coding.

תחילה נבדוק את שכיחות האותיות :

M	5
I	4
S	2
P	2
V	1
R	2
E	1
רווח	1

נבנה עץ :



נכתוב את הקוד לכל אות :

M	1111
I	00
S	01
P	100
V	1110
R	101
E	1101
רווח	1100

ולבסוף ניתן לכתוב את המילה Mississippi river לפי הקידוד של כל אות :

111100010100010100100100001100101001110110110 1

שאלה 3: טבלאות גיבוב:

•מימוש קוד כולל התיעוד נמצא בקובץ q315_

•שימוש בחבילות: pandas

import pandas as pd

-ייבוא ספריית pandas לצורך קריאת האקסל ושמירת הנתונים.

סעיף ה'

הנתונים שהתקבלו:

	Hash Table 1	Hash Table 2	Hash Table 3	Hash Table 4	Hash Table 5	Hash Table 6
method	Chain	Quadratic	Double	Chain	Quadratic	Double
Sheet 1	1.06722	2.042016	2.05882	1.07563	2.3193277	2.08403
Sheet 2	1.0	1.0	1.0	1.0	3.29411	1.78991
Sheet 3	9.4705	14.5462	3.53781	1.01680	1.76470	1.75630

בגיליון 1 מדד היעילות הטוב ביותר התקבל עבור (1,4 .Hash Table) התמודדות עם התנגשויות בשיטת השרשור.

ההסתברות להתנגשות היא קטנה מכיוון שגודל המערך גדול ממספר המפתחות .

בגיליון 1 המפתחות הם ת.ז כך שהסיכוי להתנגשות הוא נמוך ולכן שיטת השרשור עובדת ביעילות במקרה זה.

בשאר הפונקציות מדד היעילות גובה יותר.

בגיליון 2 מדד היעילות הטוב ביותר התקבל עבור (1-3 Hash Table) בפונקציית הגיבוב של חלוקה.

פונקציית הגיבוב היא mod149 המפתחות בגיליון 2 הם מספרים בסדר עולה מ1-119,

ולכן פונקציה זו היא חד חד ערכית בטווח המספרים האלו ושזה אומר שעבור כל מפתח הערך שיתקבל הוא יחיד ואין התנגשויות.

בנוסף פונקציית הגיבוב של מכפלה בשיטת השרשור (Hash Table 4)לא קיימות התנגשויות שכן ערך מדד היעילות הוא 1. מדד היעילות עבור שאר השיטות גבוה יותר.

בגיליון 3 מדד היעילות הטוב ביותר התקבל עבור פונקציית הגיבוב של מכפלה וטיפול בהתנגשויות בשיטת השרשור. ממדדי היעילות ניתן להבין כי עבור פונקציית הגיבוב של חלוקה והתנגשות בשיטת השרשור ומיעון פתוח בדיקה ריבועית. ערך מדד היעילות עבורן גבוה מאוד, מכיוון שערכי התז מסודרים בצורה יחסית רנדומלית ובקפיצות ,ולכן מאוד לא יעיל.

סעיף ו

בגיליון 1 ערך מדד היעילות הכי נמוך הוא ב Hash Table 1 ולכן נשפר אותו.

אנחנו יודעים שככל שגודל המערך יותר גדול , הסיכוי להתנגשויות יורד.

לכן נבחר להגדיל את m ואת גודל המערך ל – 401, זהו מספר ראשוני שגדול יותר מהערך הנתון לנו.

עבור הפרמטרים החדשים הערך המשופר שהתקבל הוא : 1.0

הפרש של : 0. 06722

בגיליון 2 אין צורך לשפר את מדד היעילות מכיוון שמדד היעילות שהתקבל עבור

(Hash Table 1-4)הוא 1, וזהו מדד היעילות האידיאלי.

בגיליון 3 ערך מדד היעילות של Hash Table 4 הוא הכי נמוך ולכן נשפר אותו.

אנחנו יודעים שככל שגודל המערך יותר גדול הסיכוי להתנגשויות יורד.

נבחר להגדיל את m ואת גודל המערך ל – 163, זהו מספר ראשוני שגדול יותר מהערך הנתון לנו.

עבור הפרמטרים החדשים הערך המשופר שהתקבל הוא : 1 וזהו הערך האידיאלי עבור מדד היעילות. ישנו שיפור של 0.0168067226890756.