**מטלה-1**

* שם סטודנט: אנואר סייד אחמד
* ת.ז: 206710212
* קורס: אלגורתמים מבוזרים
* מרצה: ד"ר גיא שגיא

שאלה 1:

א\_התכונה מתאר את הסדר שבו הודעות מועברות ומתקבלות ומעובדות בין התהליכים במערכת,המטרה של Massage ordering היא להבטיח שכל הצמתים יקבלו את ההודעות בסדר נכון כדי למנוע חסר עקביות בפעולות שמתבצעות במערכת.

ב\_ חסר התנאי הסיום לאלגוריתם :הצומת שמזהה שהמזהה (UID) שלו הוא הגבוה ביותר מבין כל המזהים שעברו בטבעת (עשה סיבוב שלם בטבעת)אז הוא ה LEADER.

IF V=U (ההשלמה לקוד)

ד\_לא מקיים,הסבר:  
לאחר שהצומת מזהה את עצמו כמנהיג הוא צריך לשלוח הודעה לכל הצמתים האחרים ולעדכן שהוא המנהיג וסיום הבחירה:leader(u),כלומר ברגע שהצומת מזהה עצמו כמנהיג הוא שולח את ההודעה לשאר הצמתים וכל צומת שמקבלת את ההודעה היא מעדכנת את הסטטוס שלה status=terminate ויעביר את ההודעה לצומת השכינה.

שאלה 2:

א\_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| phase | A | b | c | d | e | f |
| Out-1 | 5 | 2 | 4 | 6 | 3 | 1 |
| 1-in-r | 5 | X | X | 6 | 3 | X |
| 1-in-l | 5 | X | 4 | 6 | X | X |
| Out-2 | 5 | X | X | 6 | X | X |
| 2-in-r | 5 | X | X | 6 | X | X |
| 2-in-l | 5 | X | X | 6 | X | X |
| Out-3 | 5 | X | X | 6 | X | X |
| 3-in-r | X | X | X | 6 | X | X |
| 3-in-l | X | X | X | 6 | X | X |
| Out-4 | X | X | X | 6 | X | X |
| 4-in-r | X | X | X | X | X | X |
| 4-in-l | X | X | X | X | X | X |

ב\_מספר הסבבים הנדרשים באלגוריתם Hisrchberg-sinclair הוא log2​(n)

מכיוון שבכל סבב הטווח מוכפל, עד שהוא מכסה את כל הצמתים בטבעת. עבור n=6 נדרשים 3 סבבים

ג\_בכל סבב כמות הצמתים המשתתפים מצטמצמת בכחצי בממוצע,מכיוון שהצמתים עם UID הנמוך נפסלים ואינם שולחים הודעות בסבבים הבאים האלגוריתם דורש log(n) סבבים עד שנותר צומת אחת בלבד.  
בכל סבב כל צומת פעיל שולח שתי הודעות (שני הכיוונים),בתחילת האלגוריתם כל הצמתים שולחים הודעות (n צמתים), בכל סבב מספר הצמתים הפעילים פוחת אך בממוצע בכל סבב נשלחות O(n) הודעות.  
סיבכיות כוללת היא O(nlogn) .

שאלה 3:

A close-up of a graph

Description automatically generated

A close-up of a paper

Description automatically generated

-בהתחלה כל הצמתים מאותחלים עם new-info=TRUE וmax-uid=uid,בציור 2 עד 4 מראה איך הסטטוס של צומת 3 שינה את הסטטוס שלו ל FALSE ואחר מכן חזר לסטטוס TRUE ושדר לצמתים השכנים שלו.

שאלה 4:

א\_ מבנה הנתונים המוחזק בכל צומת:

- מחרוזת בינארית כך לכל צומת יש מחרוזת בינארית באורך m : string[m] .

- מערך שמייצג את הביטים שהתקבלו מהצמתים האחרים : received\_bits[m], בכל סיבוב מתווסף ביט חדש מהשכן עד שהמחרוזת המלאה מצומת מרוחק מתקבלת.

-הצומת מחזיקה בנוסף אינדקס (current\_bit\_index) המייצג את הביט הנוכחי במחרוזת שאותו יש לשלוח,בכל סיבוב הערך מתעדכן לשליחת הביט בהא במחרוזת(+1).

-משתנה בוליאני flag מתחיל ב TRUE, משתנה ך FALSE אם מתגלה שישי אי התאמה בין המחרוזת של הצומת לביןהביטים שהתקבלו.

-הודעה “mismach” אם הדגל הופף ל FALSE ההודעה נשלחה לצמתים כדי להפסיק בדיקת המחרוזת.

-מונה סיסובים של האלגוריתם המתחיל ב 0 ומתעדכן בכל סיבוב עד שיגיע ל m+Diam סיבובים.(round\_counter)

ב\_ מבנה ההודעה הנשלחת:

ההודעה מכילה את הביט הנוכחי מתוך המחרוזת של הצומת בהתאם לערך של current\_bit\_index.

שלבים של האלגוריתם:

1.אתחול:  
כל צומת בגרף מתחיל עם המידע האישי שלו:  
string[m], received\_bits[m](בהתחלה הוא ריק),flag=TRUE , round\_counter=0.

קוד:

current\_bit\_index = 0

round\_counter = 0

flag = TRUE

received\_bits = []

2. כל צומת שולח ביט אחד מתוך המחרוזת שלו לשכן בכיוון מוגדר (נניח, בכיוון השעון).

הביט שנשלח נקבע לפי האינדקס הנוכחי של הביט במחרוזת(current\_bit\_ index).

קוד:  
send(string[current\_bit\_index])

3. כל צומת מקבל ביט אחד מהשכן שלו,והצומת שומר את הביט שהתקבל במערך received\_bits[m].

קוד:

received\_bits.append(received\_bit)

4. הצומת משווה את הביט שהתקבל מהמחרוזת של השכן לביט המקביל במחרוזת שלו, אם הביט שהתקבל שונה, הצומת מסמן את הדגל שלו כ-.FALSE

קוד:  
flag = FALSE

5. הצומת מעדכן את האינדקס שלו לביט הבא במחרוזת (לשליחה בסיבוב הבא). הצומת מגדיל את מונה הסיבובים שלו ב-1.

קוד:

current\_bit\_index++

round\_counter++

6. האלגוריתם ממשיך עד שכולם שולחים ומקבלים את כל m הביטים שלהם,וגם עד שמידע כבר לכל הצמתים בגרף (m+Diam סיבובים),אם בדגל (flag) של הצומת נשאר TRUE אז כל המחרוזות בגרף זהות,אם הדגל של צומת מסוים היא FALSE אז ישלח הודעה “mismach” לשכניו,הצמתים מפסיקים לבדוק את הביטים הנוספים במחרוזת שלהם ומשנה את הדגל שלו לFALSE.

ג\_סיבכיות זמן:

-במקרה שכל צמתים זהים:

במקרה זה האלגוריתם נדרש להעביר כל המחרוזות ולבדוק התאמה לכן סיבכיות הזמן היא: O(m+Diam).

-במקרה שיש צומת עם מחרוזת שונה:

האלגוריתם שולח הודעה “mismach” לכל הצמתים בגרף,הודעה זו צריכה Diam מחזורים כדי להגיע לכל הצמתים וצמתים עוצרים בלי צורך לבדוק את כל ה m ביטים לכן סיבכיות הזמן היא: O(Diam).

ד\_סיבכיות תקשורת:

-במקרה שכל המחרוזות שוות:

כל הביטים של המחרוזות חייבים לעבור בין כל הצמתים לכן האלגוריתם ירוץ עד סיומו m+Diam מחזורים,בכל מחזור כל צומת שולח ביט אחד דרך כל אחת מהקשתות היוצאות ממנו כלומר |E| קשתות,לכן סיבוכיות התקשורת היא: (m+Diam)\*|E| .

-במקרה שיש מחרוזת שונה:

ברגע שצומת מגלה אי התאמה היא שולחת הודעה ”mismach”, ההודעה מתפשטת לארוך כל הקשתות תוך Diam מחזורים לכל היותר והצמתים מפסיקים להעביר ביטים יותר לכן סיבכיות התקשורת היא:Diam\*|E| .