

## אלגוריתמים מבוזרים לרשתות מחשבים

משקל: 15%

תאריך הגשה: 15/3/2023

במטלה זו נניח כי המודל סינכרוני ולא קורות בו תקלות. בחלק א' אין רעשים, כלומר כל הקודקודים יכולים לשדר לשכניהם במקביל וכל ההודעות יתקבלו בצורה תקינה. בחלק ב' נעסוק במודל יותר מציאותי בו ישנם רעשים – מודל SINR.

השתדלו לממש את האלגוריתם בצורה יעילה ככל הניתן מבחינת כל הפרמטרים. בשביל לקבל את מלוא הנקודות, הסיבוכיות לא צריכה להיפגע שלא לצורך.

### חלק א' – רשתות ללא הפרעות (90%)

נתונה רשת, המיוצגת באמצעות גרף  $G = (V, E)$ , בה קודקוד אחד בדיוק הוא שרת ושאר הקודקודים הם לקוחות. מטרתנו היא לאפשר ללקוחות להעביר הודעות לשרת עם משימות לביצוע ולשרת להחזיר תשובות ללקוחות. ברשת זו יש עלות להקמת כל קו תקשורת. עבור קוים שונים יכולה להיות עלות שונה, אשר מתבטאת במספר שלם חיובי כלשהו עבור כל קו. כלומר, מדובר בגרף ממושקל. ברצוננו להקים תשתית תקשורת (תת-גרף) ברשת, כך שתאפשר העברת הודעות בין כל הקודקודים אל השרת ובחזרה, כך שהעלות של תשתית התקשורת קטנה ככל הניתן. לצורך כך, יש לחשב עץ פורש מינימלי בגרף ולאחר מכן להעביר הודעות מהקודקודים אל השרת ובחזרה על פני צלעות העץ בלבד.

עבור היעד הנ"ל, יש לממש מספר מרכיבים:

(יש לקרוא על כל המרכיבים המתוארים בהמשך מסמך זה, לפני תחילת המימוש. זאת מכיוון שתכנון מרכיב מסוים יכול להיות מושפע גם מדרישות של מרכיב אחר.)

מרכיב 1 – מימוש אלגוריתם מבוזר לעץ פורש מינימלי של GHS – Gallager, Humblet and Spira

יש לממש את הצעדים הבאים של האלגוריתם:

בסיבוב הראשון כל קודקוד בוחר את הצלע הקלה ביותר המחוברת אליו ומודיע עליה לשכניו.

לאחר צעד זה נוצר אוסף של עצים בגרף שהם פרגמנטים של העץ הפורש המינימלי. כל קודקוד בכל עץ יודע מיהם ילדיו, על פי המידע שקיבל משכניו. כזכור, קודקוד שבחר צלע הוא הבן של הקצה השני של הצלע (פרט אולי לצלע של השורש הכפול בפרגמנט). לאחר מכן, יש לחבר את הפרגמנטים בשלבים, עד שנשאר פרגמנט אחד שהינו העץ הפורש המינימלי של הגרף כולו. כל שלב מתבצע במקביל בכל הפרגמנטים ומתואר באמצעות סדרת הצעדים הבאים עבור פרגמנט:

1. בחירת מנהיג של פרגמנט: קודקוד המגלה שהצלע שבחר נבחרה גם על ידי השכן על אותה הצלע הוא מועמד להיות מנהיג. בפרגמנט יהיו בדיוק שני מועמדים והמנהיג נבחר על פי מספר הזיהוי הגדול יותר מבין השניים.
2. הפצת מספר הזיהוי של המנהיג לכל קודקודי הפרגמנט, על פני צלעות עץ הפרגמנט, החל מהשורש.  
(לצעד זה אפשר להקצות  $n$  סיבובים)
3. כל קודקוד שולח את מספר הזיהוי של הפרגמנט שלו לכל שכניו.
4. כל קודקוד מחשב את הצלע הקלה ביותר מבין הצלעות עליהן נמצא ואשר מחברות אותו אל צלעות בפרגמנטים אחרים. שימו לב כי לא בהכרח קיימת צלע כזו בכל קודקוד.
5. לגבי הצלעות שחושבו בצעד 4, מבצעים convergencast לשורש הפרגמנט, על מנת ששורש הפרגמנט ידע את הצלע היוצאת הקלה ביותר בפרגמנט. שימו לב כי אם קודקוד מסוים בעץ מקבל במקביל מספר צלעות, אז אין צורך להעביר את כולן בכיוון השורש, אלא רק את הצלע מבניהן עם המשקל הקטן ביותר.
6. שורש הפרגמנט מבצע broadcast של משקל הצלע הקלה ביותר לכל קודקודי הפרגמנט.
7. הקודקוד בפרגמנט שמגלה שהוא על הצלע היוצאת הקלה ביותר, הופך להיות השורש החדש בפרגמנט. נסמן קודקוד זה ב- $r$ . כעת יש לשנות כיווני צלעות, בכיוון השורש החדש. לשם כך יש להפוך את כיווני הצלעות על המסלול מ- $r$  אל השורש הקודם בפרגמנט.  
(ניתן להקצות לצעד זה  $n$  סיבובים, בגלל אורך המסלול במקרה הגרוע.)
8. הקודקוד  $r$  מבקש להתחבר לפרגמנט האחר בקצה השני של הצלע היוצאת הקלה ביותר של  $r$ . לשם כך, שולח הודעה מתאימה על פני הצלע.  
בעקבות חיבורים אלה, עצים מתאחדים. אם עוד לא התקבל עץ פורש של כל הגרף (כלומר, עוד לא התבצעו  $\log n$  שלבים), נחזור לצעד 1, לביצוע השלב הבא.

## מרכיב 2: תקשורת בין השרת והלקוחות על פני עץ פורש מינימלי

בהנתן פרמטר  $n$ , יש לייצר רשת עם  $n$  קודקודים, ומכל קודקוד לבחור 7 קודקודים אקראיים ברשת, אליהם יש לחבר את הקודקוד. בשלב זה נוצר גרף לא מכוון ולא ממושקל. כעת, יש להגריל משקל חיובי שלם לכל צלע, באופן בלתי תלוי באחרים, מהטווח  $[1, \dots, 1000000000]$ . זהו טווח עם מיליארד מספרים.

יש לאפשר למשתמש לבחור את אחד הקודקודים בגרף בתור שרת. לאחר מכן, להריץ חישוב עץ פורש מינימלי, באמצעות מרכיב 1 שמימשותם. בהמשך, להפוך את השרת לשורש החדש בעץ ולהפוך כיווני צלעות מסוימות, כך שכל המסלולים בעץ יובילו אל השרת.

כעת, יש לאפשר משלוח הודעות מלקוח הנבחר על ידי המשתמש, אל השרת ובחזרה.

המשתמש יוכל ללחוץ על קודקוד בגרף ולהקליד הודעה - msg. למשל: "Hello".

ההודעה תעבור על פני צלעות העץ, בכיוון השורש – השרת. בכל מעבר, יש להוסיף להודעה את מספר הזיהוי של הקודקוד עליו ההודעה עברה, על מנת שאפשר יהיה לדעת את מסלול החזרה, כאשר ההודעה תגיע לשורש.

אז, השורש מקבל את תוכן ההודעה המקורית והמסלול עליה הגיעה. עליו לענות כך:

Your message: [msg] is received successfully, and can now be processed

כלומר, בדוגמה הנ"ל

Your message: [Hello] is received successfully, and can now be processed

הודעה זו תישלח אל הלקוח אשר ממנו התקבלה ההודעה msg.

יש צורך לתמוך בממשק משתמש גרפי בו המשתמש בוחר את הלקוח, מקליד את ההודעה ורואה כיצד ההודעה עוברת אל השרת ובחזרה. לשם כך, אפשר לשנות את הצבעים של הצלעות על פני המסלול, בזמן המעבר של ההודעה.

יש צורך גם לאפשר קבלת מידע ויזואלי לגבי משקלי כל הצלעות בגרף, בכל נקודת זמן לאחר שהוגרלו.

### מרכיב 3: בחינת המימוש על גרפים שונים

לאחר מימוש האלגוריתם עליכם לבחון את הביצועים שלו, באמצעות הפעלתו על גרפים שונים. הריצו את האלגוריתם על 20 גרפים שונים בגדלים בין 200 ל- 2000 קודקודים. לכל הרצה, יש צורך לבדוק את סכום משקלי צלעות הגרף וסכום משקלי צלעות העץ הפורש המינימלי. לשם כך, הוסיפו מרכיב שיכול לחשב סכומים אלה. **מרכיב זה בלבד** ניתן לממש בצורה לא מבוזרת. את התוצאות יש לסכם באמצעות טבלה עם 20 כניסות, כך שבכל כניסה המתארת הרצה מסוימת יופיעו מספר הקודקודים בגרף עליו הורץ האלגוריתם, סכום המשקלים של כל צלעות הגרף וסכום המשקלים של העץ הפורש המינימלי שלו.

### **חלק ב' – רשתות SINR (10%)**

הסבירו בפירוט כיצד יש לשנות את הפתרון שלכם על מנת שיתמוך ברשתות SINR. הסבירו כיצד השינוי ישפיע על מספר הסיבובים של האלגוריתם. **שימו לב:** אין צורך לממש את השינויים, רק להסביר בצורה מפורטת כיצד לעשות זאת.

### הנחיות כלליות

הקבצים שיש להגיש:

- קוד המקור של המימוש שלכם.
- מסמך המסביר כיצד להריץ וכיצד להפעיל חלקים שונים של האלגוריתם, כולל צילומי מסך.
- מסמך תיעוד: עיצוב החלקים השונים ונימוקים לבחירת מרכיבי הקוד העיקריים בפתרון. כמו כן, המסמך יכלול את הטבלה המסכמת את ביצועי האלגוריתם.
- תשובה לשאלה של חלק ב'.