אלגוריתמים מבוזרים - הרצאה 14, שעת קבלה ותרגילים

שרון מלטר, אתגר 17

18 בספטמבר 2024

תוכן עניינים

1 מציאת שידוך מקסימלי, עם דרגת יציאה 1 בגרף מכוון

1.1 הבעיה

יהי גרף מכוון G שבו דרגת **היציאה** היא 1. (שימו לב שזה לא אומר כלום על Δ)
נרצה אלג' מבוזר שמחשב שידוך מקסימלי, ב־ $O(log^*n)$ סיבובים במודל ה־ $CONGEST$.

1.2 פתרון

תחילה, נבנה גרף קשתות $L(G)$. בגרף קשתות מכוון, כל הקשתות מהוות צמתים, וקיימת בו קשת (e_1, e_2) רק אם קיים מסלול מכוון מ־ e_1 ל־ e_2 ב־ G, שהינו באורך 1.
כלומר, רק אם $e_1 = (u, v),\ \ e_2 = (v, w)$.
מאחר שב־ G דרגת היציאה של כל צומת היא 1, לכל e_1 קיימת רק קשת אחת e_2 שהוא יכול להצביע עליה. אזי גם ב־ $L(G)$, גרף הקשתות, דרגת היציאה היא 1.

באחד מתרגילי הבית ראינו שניתן לחשב 6־צביעה של גרף ב־ $O(log^*n)$ **סיבובים ע"י** אלגוריתם $Cole - Vishkin$.
לכן ניתן לחשב $O(1)$־צביעה בגרף הקשתות.
איך ניעזר בכך בשביל שידוך מקסימלי? ־ בדיוק כמו בתרגיל בית 2 שאלה 3.
נעבור על הצבעים, ובצבע ה־ i, נוסיף את כל הקשתות שצבועות ב־ i לשידוך ונמחק את שכניהם.

1.3 הוכחה שהאלג' במודל ה־ $CONGEST$

נשים לב שההודעות ששולחים באלג' הן קצרות;
ב־ $Cole - Vishkin$ רק צריך לדעת את הצבע הנוכחי של ההורה ואחר כך כל קשת צריכה לדעת את הצבע שלה ומי השכנות שלה.

בשביל סמלוץ הרצת $Cole - Vishkin$ ב־ $L(G)$ אפשר למשל לתת לכל צומת לסמלץ את הקשת האחת שיוצאת ממנו. כך צריך לקבל רק הודעות משכנים ישירים כדי לסמלץ את האלג' על גרף הקשתות. בסך הכל נצטרך רק הודעות קצרות בכל סיבוב.

1.4 הרחבת הרעיון ־ סימלוץ צביעה בגרף הרכיבים

ניתן להרחיב את רעיון הסימלוץ ולסמלץ אלגוריתם **בגרף הרכיבים**, עם דרגת יציאה 1 (נכריח אותו להיות בעל דרגת יציאה 1 בכך שנשמור רק את הקשת בעלת המזהה המינימלי מכל רכיב)

רעיון: כל רכיב צריך לדעת מידע פשוט. למשל־ את הצבע של הקשת שיוצאת מרכיב ההורה.
אפשר ללמוד בזמן שתוי בקוטר הרכיב, אם למשל הקוטר הוא $\sqrt{n}$ נוכל לחשב שידוך בגרף ב־ $\sqrt{n}log^*n$ סיבובים.

המשך הרעיון: נחשב ND ונחשב צביעה אופטימלית בכל קלאסטר בנפרד. נתייחס לצבע של כל צומת כזוג; $(C_{cluster}, C_v)$, צבע הקלאסטר וצבע הצומת v.
יש $O(logn)$ צבעים לקלאסטרים ובכל קלאסטר חישבנו צביעה אופטימלית, אז מספר הצבעים הוא לכל היותר $X(G)$ כך שסך הכל מדובר ב־ $O(logn) \cdot X(G)$ צבעים.

נשים לב שכל שני צבעים שכנים הם או באותו קלאסטר ואז הצבע שלהם שונה כי חישבנו צביעה חוקית בתוך הקלאסטרים. אם הם בקלאסטרים סמוכים, אז הצבע של הקלאסטרים יהיה שונה (כלומר ה־ $C_{cluster}$ שלהם יהיה שונה)

בתרגיל הבית, הרבה פתרונות היו בכיוון, אבל לא בחרו את **הצבע הסופי** נכון. נעבור לרדת על הדרכים השגויות שהוצגו;

- שרשור $C_{cluster}C_v$ (שרשור בינארי) לא עובד כי שרשור של זוגות שונים יכול להוביל לאותו הצבע.

- גם הקצאה של $logn$ ביטים לכל אחד מהם בעייתית; $c|\frac{C_{cluster}}{loglogn}$ מכיוון שאז הצבעים מטווח פולינומי אבל אולי $X(G)$ קבוע והגדלנו את מס' הצבעים יותר מדי (בפקטור פולינומי)

2 שאלה ממבחן - מועד ב' 3202

2.1 הבעיה

נרצה למצוא $(deg+1)$-צביעה כך שכל צומת יקבל צבע שתלוי בדרגה של. למשל, אם v מדרגה d_v נרצה לתת לו צבע מ- $\{1, 2, ..., d_v, d_v + 1\}$.

2.2 סעיף א' - הוכחת קיום הצביעה

הוכיחו שתמיד קיימת צביעה כפי שתוארה.

נעשה זאת בכך שנוכיח שקיים אלג' סדרתי שפותר את הבעיה. שימו לב שהוא לא צריך להיות יעיל, רק נכון.
אפשר לסדר את הצמתים באיזשהו סדר, ולעבור עליהם לפיו. כשהגענו לצומת v, נבחר לו צבע ששונה מהצבעים של שכניו שכבר נצבעו. מאחר שיש לו d_v שכנים, והם לכל היותר צבועים ב- d_v צבעים שונים, קיים צבע מ- $\{1, 2, ..., d_v, d_v + 1\}$ שאף שכן שלו לא צבוע בו.

2.3 סעיף ב' - $(deg+1)$-צביעה ב- $O(\Delta + log^*n)$ סיבובים במודל ה- $LOCAL$

קל להבין מה אנו נדרשים לעשות מהכותרת (תרגיל)

נתחיל מלחשב $(\Delta+1)$-צביעה. ניתן לעשות זאת ב- $O(\Delta + log^*n)$ סיבובים.
לאחר מכן, נפעל באיטרציות $i = \Delta + 1, \Delta, ..., 1$.
באיטרציה ה- i כלל הצמתים שצבועים בצבע ה- i יבחרו צבע חדש שהוא הצבע המינימלי הפנוי שלא תפוס ע"י שכנים שלהם.

כמו בסעיף הקודם, הצבע החדש שכל צומת יבחר יהיה בין $1, 2, ..., d_v + 1$ כי יש לו רק d_v שכנים. בנוסף, הצביעה נשארת חוקית:
התחלנו עם צביעה חוקית, ובכל איטרציה אנחנו פועלים רק בצמתים שהם בעלי אותו צבע i כך שהם **לא שכנים** והם בוחרים צבע ששונה משכניהם כך שאחרי כל איטרציה הצביעה נשארת חוקית. בסיום, עברנו על כל הצמתים, ולכן קיבלנו $(deg+1)$-צביעה חוקית.

3 חזרה - אלג' $Boruvka$

בהתחלה כל צומת הוא רכיב קשירות.
בכל איטרציה;

1. מוצאים קשת מינימלית יוצאת מכל רכיב.
2. מאחדים בין הרכיב שממנו היא יוצאת לרכיב אליו היא מכוונת, כך שהם רכיב קשירות יחיד.

חוזרים על כך עד שנותר רק רכיב קשירות אחד.
נשים לב שבכל איטרציה מס' רכיבי הקשירות קטן לפחות פי 2 (במקרה בו כל רכיב בוחר להתאחד עם רכיב שבוחר להתאחד גם איתו) כך שיש לכל היותר $O(logn)$ איטרציות.

איחוד הרכיבים יכול להיעשות בזמן שלוקח למצוא את קוטר הרכיבים (כי כל צומת צריך לדעת מהו המזהה החדש שנבחר, והוא המזהה המינימלי שברכיב) לכן ניתן לממש את האלג' ב- $O(nlogn)$.

<u>מטרה:</u> להקטין את הסיבוכיות ל- $\widetilde{O}(D + \sqrt{n})$.

רעיון: נפריד את הרכיבים לקטנים וגדולים; קטנים הינם רכיבים בגודל עד $\sqrt{n}$ והשאר הם רכיבים גדולים.
(שימו לב שיש רק $O(\sqrt{n})$ רכיבים גדולים)

ברכיב קטן, אפשר למשל למצוא קשת מינימלית יוצאת ב־ $O(\sqrt{n})$ סיבובים מכיוון שזהו חסם עליון לקוטר שלהם.

ברכיב קטן, אפשר לתקשר על עץ BFS של הגרף G כדי למצוא קשת מינימלית. ניתן לעשות זאת עבור **כל** $\sqrt{n}$ הרכיבים הגדולים ב־ $\tilde{O}(D+\sqrt{n})$ סיבובים ע"י *pipelining*.