אלגוריתמים מבוזרים - הרצאה 4, תרגילים נבחרים

שרון מלטר, אתגר 17

21 בספטמבר 2024

תוכן עניינים

1 חסמים תחתונים לצביעת עצים לא מכוונים

ראינו כל מיני דוגמאות של גרפים בהם ניתן לחשב צביעה ב־ $O(log*n)$ וגם בעצים לא מכוונים ניתן לחשב צביעה ב־ $O(log*n)$ ע"י הרחבת האלגוריתם לטבעת.
אבחנה: עץ תמיד ניתן לצבוע בשני צבעים. איך? כל רמה זוגית בצבע אחד, וכל רמה אי־זוגית בצבע שני. אך לא מדובר ברעיון הכי יעיל. אבל, אם ידועה לנו האוריינטציה (כיווני הקשתות) של הגרף, נוכל לחשב צביעה שלו ב־ $O(log*n)$ **סיבובים**.

<u>הגדרה:</u> $girth$ (מותן) של גרף הוא אורך המעגל הקצר ביותר בגרף.

<u>משפט $Erolos$:</u> לכל n גדול מספיק, קיים גרף G_n^* בעל n צמתים עם $girth \leqslant \frac{logn}{4loglogn}$ ומס' צביעה $\frac{logn}{4loglogn} \leqslant X(G_n^*)$

<u>טענה:</u> כל אלגוריתם שצובע **עץ לא מכוון** בפחות מ־ $\frac{logn}{4loglogn}$ צבעים, מצריך $\Theta(\frac{logn}{loglogn})$ סיבובים.

<u>הוכחת הטענה:</u>

<u>רעיון:</u> אלגוריתם בלוקל הוא פונקציה שמסתכלת ע"פ סביבה ובוחרת צבע. אם יש לי גרפים שונים אבל שיש להם סביבות זהות, אז האלגוריתם יפעל באותו אופן (כפי שלמדנו אתמול)

נניח בשלילה שקיים אלגוריתם שצובע עץ לא מכוון בפחות מ־ $\frac{logn}{4loglogn}$ צבעים ולוקח לכל היותר $r \leqslant \frac{logn}{4loglogn}$ סיבובים ונראה שהאלגוריתם הנ"ל יכול לשמש למציאת צביעה חוקית ב־ G_n^*.

כיוון שהמעגל הקצר ביותר ב־ G_n^* הוא באורך לפחות $t = \frac{logn}{4loglogn}$. סביבת $\frac{t}{2} - 1$ של כל הצמתים נראית כמו עץ מכיוון שאין בה מעגלים (מאחר שלא קיים מעגל באורך הקטן מ־ t) ולמה זה טוב? ־ מכיוון שהסביבות האלה חסרות מעגלים, ניתן להשתמש באלגוריתם שלמדנו לצביעת עצים בשביל לצבוע את G_n^*. כל צומת ילמד את סביבת r שלו, כאשר $r \leqslant \frac{t}{2} - 2$ ויבחר צבע לפי האלגוריתם, שלוקח r סיבובים.

נטען שצמתים שכנים תמיד יבחרו צבעים שונים. כיצד נוכיח זאת? ־ יהיו שני צמתים שכנים u,v כך ששניהם למדו את הסביבות r שלהם. נסתכל על הגרף שמכיל את הסביבות r של u,v ונסמנו $T_{u,v}$. מכיוון שכל הצמתים בו במרחק לכל היותר $r+1$ מ־ u $(r+1 \leqslant \frac{t}{2} - 1)$ וכיוון שהאלגוריתם מחשב צביעה חוקית לעצים, בפרט היא מחשבת צביעה חוקית עבור העץ $T_{u,v}$, לכן u,v יבחרו צבעים שונים. אזי קיבלנו צביעה חוקית ב־ G_n^* עם פחות מ־ $\frac{logn}{4loglogn}$ צבעים, **בסתירה** לכך ש־ $\frac{logn}{4loglogn} \leqslant X(G_n^*)$. לכן מספר הסיבובים לא יכול להיות קטן מ־ $\Theta(\frac{logn}{4loglogn})$, כנדרש.

2 צביעת עץ מכוון, כאשר כל הקשתות מכוונות לכיוון השורש

שימו לב שבעץ רלוונטי דרגת היציאה של כל צומת היא לכל היותר 1.

<u>טענה:</u> ניתן לצבוע 6־צביעה של עץ מכוון ב־ $O(log*n)$ סיבובים.
מדובר בהוכחה קלה (והיא דומה לשאלה מתרגיל הבית הראשון) לכן זהו תרגיל לסטודנט.

בהינתן 6־צביעה של עץ מכוון, נרצה לראות כיצד ניתן לקבל 3־צביעה ב־ $O(1)$ סיבובים נוספים. (כך שניתן לקבל 6־צביעה ב־ $O(log*n)$ סיבובים)

שימו ♡ ־ הדרגות בעץ לא חסומות, אז אם ניתן לכל הצמתים עם צבע מקסימלי לבחור צבע חדש במקביל, לא מובטח שיבחרו צבע מ־ $\{1,2,3\}$ כיוון שיכולים להיות להם יותר מ־ 2 שכנים.

2.1 הוכחה

<u>טענת עזר:</u> בהינתן 6־צביעה של עץ מכוון, ניתן **בסיבוב אחד** לחשב 6־צביעה עם התכונה הנוספת שלכל צומת כל השכנים צבועים בשני צבעים לכל היותר.

<u>הוכחה:</u> בהינתן 6־צביעה, נחשב 6־צביעה חדשה באופן הבא:
כל צומת ייקח את הצבע של ההורה שלו בעץ. השורש ייקח צבע חדש ששונה מהצבע הקודם שלו. נשים לב שבצביעה החדשה עבור כל צומת, כל השכנים שלו צבועים בלכל היותר 2 צבעים: כל הילדים שלו לקחו את הצבע שלו (צבע אחד) וההורה שלו צבוע בצבע אחר (צבע שני).

נותר להראות שמדובר ב־ 6־צביעה חוקית.
יהיו u,v צמתים שכנים ונניח בה"כ ש־ u היא ההורה של v. נסמן ב־ w את ההורה של u, אם הוא קיים ונסמן ב־ $C(v)$, $C(u)$ $C(w)$ את הצבעים המקוריים של u,v,w. הצבע החדש של u הוא $C(w)$ והצבע החדש של v הוא $C(u)$. מאחר ש־ u,v שכנים, מתקיים $C(w) \neq C(u)$.
אם u הוא השורש, אז הצבע החדש של v הוא $C(u)$ ו־ u בחר צבע אחר, לכן גם כאן הצבעים שונים וקיבלנו שהצביעה חוקית :)

בעזרת הטענה, בהינתן 6־צביעה, נחשב 3־צביעה באופן הבא;
עבור $i \in \{4,5,6\}$:

1. נפעיל את טענת העזר.
2. ניתן לכל הצמתים בעלי צבע i לבחור צבע חדש במקביל־ הצבע המינימלי שלא תפוס ע"י אחד השכנים שלהם.

כיוון שאנחנו מפעילים את טענת העזר, כשנגיע לשלב השני השכנים של כל צומת יהיו צבועים בלכל היותר 2 צבעים שונים, ולכן מובטח שכל צומת ייבחר צבע מ־ $\{1,2,3\}$ ובסיום האלגוריתם נקבל 3־צביעה. הצביעה חוקית מכיוון שבשלב השני אף צומת לא בוחר צבע ששייך לאחד משכניו :)

3 צביעה עם פחות צבעים בסיבוב אחד

נוכיח כי בהינתן Δ^3-צביעה ניתן בסיבוב אחד למצוא $O(\delta^2)$-צביעה.

<u>תזכורת:</u> בצביעה ע"י $\Delta-cover-free\ families$ נרצה לתת לכל צבע $1 \leqslant i \leqslant \Delta$ קבוצה $S_i \subseteq \{1,2,...,O(\Delta^2)\}$ כך שכל צומת v שצבוע בצבע מקורי i, יוכל לבחור צבע מתוך הקבוצה S_i שלא נמצא בקבוצות S_j של אף אחד מהשכנים שלו.
פורמלית, לכל $\Delta+1$ אינדקסים $i_0,i_1,...,i_\Delta$ נרצה ש- $S_{i_0}/\cup_{j=1}^{\Delta} S_{i_j} \neq \varnothing$ (כלומר שב- S_{i_0} יהיה איבר שלא קיים ב- $\cup_{j=1}^{\Delta} S_{i_j}$).

נרצה דרך חלופית לבנות $\Delta-cover-free\ families$ שמתאימה לדרישות השאלה ע"י **פולינומים**.
(אפשר להשתמש ברעיונות דומים כדי להראות בניה עם פרמטרים דומים לאלו שראינו בהרצאה)

<u>רעיון:</u> נבחר מס' ראשוני q כך ש- $2\Delta \leqslant q \leqslant 6\Delta$. המטרות הן:

1. להתאים לכל צבע i פולינום P_i מעל השדה F_1 כך שצבעים שונים יקבלו פולינומים שונים.
2. לכל פולינום P_i להתאים קבוצה $S_i \subseteq \{1,2,...,O(\Delta^2)\}$ כך שאוסף הקבוצות יהיה $\Delta-cover-free$.

<u>פיתרון:</u>

1. <u>תזכורת:</u> התחלנו מ- Δ^3-צביעה. נשים לב שאם נחבר פולינומים מדרגה לכל היותר 2 הם מהצורה ax^2+bx+c.
יש q אפשרויות לבחור כל אחד מ- a,b,c כיוון ש- $3\Delta \leqslant q$ נקבל שיש יותר מ- Δ^3 פולינומים כאלו, אז ניתן להתאים לכל צבע פולינום כנדרש.
למשל, הצבע ה- 1 יתאים לבחירה $a=b=c=0$ וכן הלאה. כל הצמתים יכולים לחשב את ההתאמה הלוקלית ללא תקשורת.

2. עבור הפולינום P_i נגדיר את הקבוצה:

$$S_i = \{(x, P_i()|0 \leqslant x \leqslant q-1)\}$$

כלומר נציב בפולינום P_i את הערך x לכל x אפשרי וניקח את הנקודות $x, P_i(x)$.
נשים לב ש- $q=O(\Delta)$ וגם $A(x)$ הוא ערך בשדה ולכן גם $P_i(x)=O(\Delta)$ לכן נקבל שיש $O(\Delta^2)$ ערכים אפשריים מהצורה $x, P_i(x)$ ועבור כל x ו- i אפשרים ניתן להתאימם לכל ערך כזה מספר בין 1 ל- $O(\Delta^2)$ ונקבל ש- $S_i \subseteq \{1,...,O(\Delta^2)\}$
(כל הקבוצה בנפרד מכילה $q=O(\Delta)$ זוגות)

נשאר להראות שהקבוצות הן $\Delta-cover-free$.
נסתכל על קבוצות שונות $S_i \neq S_j$ ונניח שקיים זוג כך ש- $(x,P_i(x))=(y,P_j(y))$ (זוג משותף ל- S_i,S_j)
אז נקבל ש- $P_i(x)=P_j(x),\ x=y$ כלומר מצאנו נק' חיתוך בין הפולינומים. ידוע שפולינומים שדרגתם היא לפחות 2 נחתכים לכל היותר ב-2 נקודות, ולכן לכל זוג הקבוצות S_i,S_j נחתכות לכל היותר ב- 2. בפרט;

$$|S_0/\cup_{j=1}^{\Delta} S_{i_j}| \geqslant q-2\Delta > 0$$

זאת מכיוון שלפי בחירת q מתקיים $2\Delta < 2\Delta \leqslant q$. קיבלנו שהחיתוך לא ריק, לכן הקבוצות הן $\Delta-Cover-Free$.

נשים לב שבשביל האלגוריתם כל צומת צריך לדעת רק את הצבעים המקוריים של השכנים שלו. כלומר, אנו צריכים רק סיבוב אחד כדי לדעת מה הקבוצות S_j שמתאימות להם (ההתאמה בין כל צבע j לקבוצה S_j ידועה לכל הצמתים כפי שהוסבר קודם)
ראינו אלגוריתמים לצביעה בסיבוכיות $O(\Delta+log*n)$. מדובר בסיבוכיות טובה עבור Δ קטן. עבור Δ גדול יותר, יש אלגוריתמים יעילים יותר אשר לוקחים $O(logn)$ סיבובים. נבנה אלגוריתמים אלה בתרגיל הבא :)

4 אלגוריתם ל־ 2Δ־צביעה ב־ $O(logn)$ סיבובים

נמצא אלגוריתם המתאים ל־ Δ גדול.
נבנה אלגוריתם רנדומי שמחשב 2Δ־צביעה ב־ $O(logn)$ סיבובים ב־ $CoNGEST$.

4.1 האלגוריתם

בכל סיבוב כל צומת יבחר באקראי צבע מתוך $\{1,2,...,2\Delta\}$. אם בחר צבע שאף אחד מהשכנים שלו לא בחר, ייצבע בצבע זה וישאר בו.
בכל סיבוב לכל צומת יש סיכוי של לפחות $\frac{1}{2}$ להיצבע. השכנים שלו בחרו ב־ Δ צבעים לכל היותר, ולכן אם u בחר באחד מ־ Δ הצבעים האחרים, יצליח. אם נחזור על כך $O(c \cdot logn)$ סיבובים, הסיכוי ש־ u לא נצבע הוא $\frac{1}{n^c} = (\frac{1}{2})^{c \cdot logn}$
הסיכוי **שקיים** צומת שלא נצבע חסום ע"י $union\ bound$ ב־ $\frac{1}{n^{c-1}} = \frac{n}{n^c}$ ולכן בהסתברות של לפחות $1 - \frac{1}{n^{c-1}}$ האלגוריתם ייספק צביעה חוקית :)

5 $deg+1$-צביעה

$deg+1$-צביעה היא צביעה שבה צומת v מדרגה d יקבל צבע מ- $\{1,2,...,d+1\}$ (ייתכן ש- $\Delta >> d$).
ניתן לפתור את הבעיה כך;

1. הראו שבכל גרף G קיימת $deg+1$-צביעה.
2. הראו אלג' מבוזר שמחשב $deg+1$-צביעה ב- $O(\Delta + log*n)$ סיבובים.

אלגוריתם חמדן סדרתי מספק את הדרישה הראשונה ובנדיבות נציג רעיון לדרישה השנייה- חישוב של $\Delta+1$-צביעה ב- $O(\Delta + log*n)$ סיבובים.