אלגוריתמים 1 ‑ הרצאה 10: שבירת סימטריה

13 בינואר 2026

גיא יער־און

0.1 מבוא והגדרת הבעיה

נניח כי יש לנו שני אנשים: אליס ובוב. שניהם נמצאים בשיחת זום, ואצל שניהם המצלמות כבויות. כל אחד מהם רוצה לומר משהו אל הצד השני. למשל: אליס רוצה לומר לבוב שקוראים לה אליס, ובדומה בוב רוצה להגיד לאליס ששמו הוא בוב. יותר מזה: יתכן שלשניהם קוראים אליס (לא בהכרח שהם בשם שונה). אם שניהם ידברו בו זמנית, הם יעלו על הקול אחד של השני ולא יצליחו לשמוע את הקול. המטרה היא להגיע למצב שרק אחת "משדרת" קול.

נשים לב כי כל אלגוריתם דטרמיניסטי לא יצליח לפתור את הבעיה. מדוע? מהו אלגוריתם דטרמיניסטי? אלגוריתם דטרמיניסטי הוא קוד כתוב שידוע מראש. ולכן אם בוב ואליס ישתמשו בקוד שנמצא במחשב שלהם בו זמנית, הוא יגיד להם לעשות אותו הדבר בדיוק. מכאן שאנחנו חייבים להשתמש ברנדומיות.

בעזרת מטבע רנדומי אצל כל אחד מהמשתתפים ניתן להצליח. נניח כי נגדיר את הטלת 1 להיות שהמשתתף מדבר ו0 שהמשתתף שותק. נראה כי במקרה של אליס ובוב מס' האפשרויות להטלת המטבע הינו:

$$00, 11, 01, 10$$

מצב טוב עבורנו הוא $01, 10$. ומכאן מה הסיכוי להצלחה ושיהיה משתתף אחד בדיוק שמדבר? $\frac{1}{2}$.

מכאן אפשר לקבל אלגוריתם: כל עוד אין הצלחה ‑ הטל מטבע, אם יוצא אחד אז תשדר.

נראה כי מס' הנסיונות עד להצלחה הראשונה הוא משתנה מקרי שמתפלג גאומטרית, ולכן תוחלת מס' הנסיונות תהיה $2 = \frac{1}{\frac{1}{2}} = \frac{1}{p}$.

ניתן גם לבדוק אחרי כמה סיבובים תהיה הצלחה בסיכוי גבוה. נראה כי כדי שלא תהיה הצלחה בk סיבובים ההסתברות הינה:

$$\frac{1}{2} \cdot \frac{1}{2} \cdot .... \cdot \frac{1}{2} = \frac{1}{2^k}$$

עבור $k = clogn$ עבור $c > 1$ קבוע, נקבל כי הסיכוי לא להצליח בk נסיונות הוא:

$$\frac{1}{2^k} = \frac{1}{2^{clogn}} = \frac{1}{n^c}$$

כלומר, הסיכוי לא להצליח קטן פולינומית. ומכאן שיש הצלחה בסיכוי די גבוה.

אפשר להרחיב את הבעיה, מה אם יש שלושה אנשים ואנו רוצים לשבור סימטריה? נרצה שכל אחד ינסה לשדר בסיכוי שליש. בשביל שתהיה הצלחה בניסוי אחד: צריך שאחד ישדר, ושניים אחרים ישתקו. הסיכוי לכך (מתפלג בינומית) הינו:

$$\binom{3}{1} \times p \times (1-p)^2 = 3 \times \frac{1}{3} \times (\frac{2}{3})^2 = \frac{4}{9}$$

ומכאן שבתוחלת לאחר $\frac{1}{\frac{4}{9}} = \frac{9}{4} = 2.25$ סיבובים תהיה הצלחה.

וכמובן איך לא, מה קורה כאשר ישנם n שחקנים? נשים לב (גם כשיש 3 שחקנים) שהשחקנים זקוקים לדעת מהו n. נניח כי כל אחד מהם מנסה לשדר בסיכוי p. הסיכוי לשידור (בחירת מנהיג) הינו:

$$Pr[Success] = \binom{n}{1} \times p \times (1-p)^{n-1}$$

נרצה למקסם את הסיכוי להצלחה, כלומר p. מכאן שהסיכוי להצלחה המקסימלי יתקבל כאשר $p = \frac{1}{n}$ (גזירה פשוטה מראה זאת)
מכאן שהסיכוי להצלחה הינו:

$$Pr[Success] = \binom{n}{1} \times \frac{1}{n} \times (1-\frac{1}{n})^{n-1} = (1-\frac{1}{n})^{n-1} = \frac{(1-\frac{1}{n})^n}{1-\frac{1}{n}} \le \frac{e^{-1}}{1-\frac{1}{n}} =$$

$$\Longrightarrow Pr[Success] \le \frac{n}{e(n-1)} \le \frac{1}{e}$$

ולכן תוחלת מס' הנסיויים עד להצלחה תהיה **בערך** $e = \frac{1}{\frac{1}{e}}$.

אם נגדיר פורמלית את הבעיה:
קלט: n שחקנים.
פלט: מנהיג יחיד.

0.2 המודל המבוזר המקומי

נניח שיש n מחשבים ברשת מחשבים, כל קודקוד ייצג מחשב וישנן קשתות בין מחשבים (גרף). כל קשת היא חיבור רשת בין מחשבים. לכל מחשב יש כוח חישוב אינסופי. כלומר - אנחנו נניח שכל מחשב באופן מקומי יכול להריץ אלגוריתמים מאוד מסובכים שזמן הריצה שלהם מאוד גבוה: בחינם. כלומר: תאורטית, כל מחשב יכול לפתוח בעיות NP קשות בשנייה. בכל יחידת זמן, כל מחשב יכול לשלוח הודעות גדולות כרצונו לכל אחד משכניו (זה יעלה סיבוב אחד של תקשורת). נראה כי בזמן שקודקוד אחד שולח הודעה לשכניו, גם שאר הקודקודים שולחים הודעה לשכניהם. כלומר: שליחת ההודעות מתבצעת במקביל.
במודל הזה, נמדוד את היעילות של האלגוריתמים שלנו באמצעות מס' סבבי התקשורת.
למשל, בהינתן גרף כנ"ל:

$$a_1 \to a_2 \to a_3 \to .... \to a_d$$

אנחנו נאלץ d לסבבי תקשורת, בסבב הראשון ההודעה תצא מa_1 אל a_2, בשלב השני מa_2 לa_3 וכן הלאה. אין לנו ברירה אחרת כאן כיוון שאין לנו דרכי קיצור.
בחלק זה של הקורס: נסמן בn את מס' הקודקודים, **ונניח כי כל מחשב יודע את n.** לכל מחשב **אין** id (יש id אם מרשים שימוש ב$random$). ופרמל:

הגדרה: המודל המפוזר המקומי הוא גרף לא מכוון $G=(V,E)$ כל קודקוד הוא מחשב שמריץ קלט מקומי. זמן מחולק לסבבים, בכל סבב כל קודקוד יכול לשלוח הודעה גדולה כרצונו לכל שכניו. כל קודקוד מכיר את שכניו. בעולם הדטרמיניסטי מניחים כי לכל קודקוד ישנו id, עם זאת במודל המפוזר לכל קודקוד אין id.
סבב הוא חישובי מקומי פולינומי, שיש בו שליחה וקבלת הודעות. זמן הריצה יהיה כמספר הסבבים.
כל קודקוד מריץ מראש את אותו האלגוריתם.

במודל המבוזר המקומי, ישנן שתי בעיות של שבירת סימטריה שיוכלות לעניין אותנו.
צביעה: המטרה היא לצבוע את קודקודי הגרף (לתת מספרים שהם צבעים) כך שלכל קשת שני הקודקודים צבועים בצבעים שונים. נשים לב שללא רנדומיות לא ניתן לפתור את הבעיה, שכן יתכן גרף סימטרי עם שני קודקודים v_1, v_2. לכל אחד מהם יש שלושה שכנים נוספים. אז מבחינת כל קודקוד יש אותו, יש לו שלושה שכנים ויש קודקוד נוסף שיש קשת בניהם שגם לו יש שלושה קודקודים. אין שוני בניהם ולכן הם יבצעו את אותה ההחלטה באלגוריתם דטרמיניסטי.

מציאת קבוצה בלתי תלויה מקסימלית: בהינתן גרף רוצה לבחור תת קבוצה של הקודקודים שהיא:
1. מקסימלית (במובן הלוקאלי: כלומר לא ניתן להוסיף עוד קודקוד ולהשאר בקבוצה בת"ל)
2. אין זוג שכנים שנבחר.
במדעי המחשב לרוב מדברים על קבוצה בלתי תלויה מקסימומית (מקסימום), זו בעיה שניתן לפתור.
כאן נרצה למצוא את הקבוצה הבלתי תלויה בגודל הכי גדול. - זו בעיה הרבה יותר קשה.

0.3 בחירת ID במודל המפוזר

כרעיון ראשוני, נניח כי כל קודקוד בוחר ID מהטווח $[n]$ באקראי, מה הסיכוי שיש התנגשות? נקבע $i\in[n]$ ונגדיר n_i כקבוצת כל הקודקודים שבחרו את i.

$$Pr[n_i=0]=\binom{n}{0}\times(1-\frac{1}{n})^n=\frac{1}{e}$$

נבחין כי הסיכוי ש$n_i=0$ גורר כי כל תא קיבל בדיוק כדור אחד, שכן נניח כי יש תא עם יותר משני כדורים אזי בהכרח יהיה תא עם אפס. מכאן שהסיכוי שאין התנגשות הוא $\frac{1}{e}$.

$$Pr[\exists_{i\in[n]}n_i=0]=Pr[\bigcup\{n_i=0\}]\leq\sum_{i=1}^{n}Pr[n_i=0]\approx\frac{n}{e}$$

ונבחין כי מדובר בחסם די גרוע. כלומר הסיכוי שיש התנגשות (תא אחד עם אפס) חסום לנו כאן ב$\frac{n}{e}$, נרצה לשפר. נבחין כי הסיכוי שקיים תא עם אפס הוא:

$$Pr[]=\frac{n^n-n!}{n^n}=1-\frac{n!}{n^n}$$

כלומר, הסיכוי הזה די גדול שכן $n! << n^n$ ולכן הסיכוי שיש id שנבחר פעמיים (התנגשות) יחסית גדול .

כעת, נבחר id באקראי מתוך טווח $Id \in [n^c]$. עבור $c > 0$ פרמטר. נבחין כי הסיכוי להתנגשות הינו:

$$Pr[Id(u) = Id(v)] = \frac{1}{n^c}$$

$$Pr[\exists u \neq v : Id(u) = Id(v)] = Pr[\bigcup_{u \neq v} Id(u) = Id(v)] \leq \sum_{u \neq v} Pr(Id(u) = Id(v)) = \binom{n}{2} \times \frac{1}{n^c} < \frac{n^2}{n^c} = \frac{1}{n^{c-2}}$$

ולכן, עבור $c \geq 3$ נקבל כי הסיכוי שלא תהיה התנגשות, גבוה מאוד.

0.4 בעיית הצביעה

קלט: גרף $G = (V, E)$.
פלט: פונקציית צביעה $C : V \to [1, ..., c]$ כך שהצביעה חוקית (לכל $(u, v) \in E$ מתקיים $c(u) \neq c(v)$) וכן c מינימלי.

נזכר כי בגרף דו צדדי ניתן תמיד לצבוע אותו בשני צבעים, גרף תלת צדדי ניתן לצביעה בשלושה צבעים. גרף n־צדדי ניתן לצביעה בn צבעים. (שכן די ברור הרעיון אין קודקודים בתוך כל צד אז אפשר לצבוע כל צד בצבעים שונים).
באופן כללי, הקושי הוא במציאת המספר הקטן ביותר של צבעים שבהם ניתן לצבוע את הגרף באופן חוקי. מדובר בבעיה מאוד קשה. לא ניתן לפתור אותה באופן דטרמיניסטי בפחות מ$O(2^n)$. ואף ־ מאמינים כי לא ניתן להגיע לזמן פולינומי. אם כן, יש משפחות של גרפים שניתן לצבוע אותם באופן פולינומי. גרף דו צדדי למשל.

נסמן בΔ את הדרגה הכי גדולה בגרף. כלומר, לכל קודקוד $v \in V$ ישנה דרגה $\Delta \leq deg(v)$.
הבחנה: ניתן לצבוע את הגרף באופן חוקי ב$\Delta + 1$ צבעים. מדוע? נתחיל בקודקוד מדרגה d כלשהי, בהכרח $d \leq \Delta$, גם אם $d = \Delta$ הוא משוייך לΔ קודקודים שכל אחד מהם תפס צבע אחר, במקרה הגרוע ביותר שאכן $d = \Delta$ עדיין הוא יוכל להשתמש בצבע האחרון. באופן כללי הוא יוכל להשתמש ב$\Delta + 1 - d \geq \Delta + 1 - \Delta = 1$ צבעים.

0.5 צביעה במודל המבוזר המקומי

נניח שיש לנו קודקוד v ויש לו שכנים $v_1, .., v_k$. הבעיה של הקודקוד v הוא שהוא לא יודע כיצד שכניו פועלים. למשל: אם קודקוד v היה יודע ששכניו אינם שכנים אחד של השני ־ אזי קודקוד v היה רוצה לומר להם: תצבעו כולכם באותו הצבע, ואני אצבע את עצמי בצבע שונה משלכם.
כרעיון בסיסי מאוד ־ אני יכול להחליט שהקודקוד v ישלח הודעה $(v_1, v_2, ..., v_k)$ לכל שכניו. כמו כן: כל שכן ישלח הודעה לכל השכנים שלו עצמו. ואז ־ אני אקבל את ההודעות של כל שכניי, ואוכל להסתכל האם באחת ההודעות אני מזהה קודקוד שכבר יש לי. כלומר: האם השכנים שלי הם גם שכנים אחד של השני. נשים לב שקצת רימינו ־ שכן טענו כי לקודקודים אין id, אז איך נוכל לשלוח הודעה שכזו? נדבר על תהליך בחירת ID שקורה במקביל עבור כל הקודקודים.

בחירת ID:

1. כל קודקוד בוחר id בין המספרים $[1,2,.....,n^{c+2}]$ עבור c קבוע.
2. שלח ID לכל השכנים. (שכל אחד ידע את ה$Id's$ של השכנים שלו)

נרצה לבדוק מה הסיכוי שישנם שני קודקודים עם אותו id.

$$Pr[sameID] = 1 \times \frac{1}{n^{c+2}} = \frac{1}{n^{c+2}}$$

נסמן בA_{uv} את המאורע שבו u וv בחרו את אותו Id. מכאן $Pr[A_{uv}] = \frac{1}{n^{c+2}}$. נסתכל על המאורע הבא, שמשמעותו שאף אחד לא בחר את אותו id.

$$Pr[\overline{\bigcup_{u,v\in V} A_{uv}}] = 1 - Pr[\bigcup_{u,v\in V} A_{uv}]$$

נראה כי

$$Pr[\bigcup_{u,v\in V} A_{uv}] \leq \sum_{u,v\in V} Pr[A_{uv}] \leq n^2 \times \frac{1}{n^{c+2}} = \frac{1}{n^c}$$

וקיבלנו כי

$$Pr[\overline{\bigcup_{u,v\in V} A_{uv}}] = 1 - Pr[\bigcup_{u,v\in V} A_{uv}] = 1 - \frac{1}{n^c}$$

ולכן הסיכוי לטעות קטן פולינומית, והסיכוי להצלחה גדול מאוד פולינומית.

סה"כ קיבלנו כי באמצעות טכניקה רנדומית פשוטה, יצרנו לכל קודקוד ID. מכאן: יש משמעות לשליחת הודעה לכל השכנים של רשימת השכנים.

נראה כי ישנו קושי - בהחלט יכול להיות שאפילו שלשכנים שלי אין קשתות ביניהם, עדיין לא ניתן לצבוע את כל השכנים באותו צבע. הרבה דוגמאות יכולות להעיד על כך: הרעיון הזה פשוט מדי, וחושב לוקלית מקומית אך הגרף הרבה יותר גדול מזה. בחירה מקומית יכולה להשפיע על הגרף כולו באופן שלא ציפינו. יתרה מזאת - כל מעגל אי זוגי דורש לפחות 3 צבעים. ושיטה זו במעגל תניב 2 צבעים. וכן: יש קושי להבין האם קודקוד נמצא במעגל אי זוגי. שכן יתכן כי גודל המעגל האי זוגי מאוד מאוד גדול ויקח המון זמן והודעות להבין שאנחנו נמצאים בכזה.

אנחנו נרצה לצבוע את הגרף באמצעות 2Δ צבעים. (ראינו כבר כי ניתן להשתמש ב$\Delta+1$ צבעים, אך המטרה באלגוריתם שנראה בהרצאה הוא לא משהו חדשני - אלא להבין את המודל המבוזר המקומי)

0.6 אלגוריתם צביעה

כעת נתאר את האלגוריתם שיצבע את הגרף באמצעות 2Δ צבעים.

1. נבחר צבע באקראי מבין $[1,...,2\Delta]$. נשים לב - ישנה כאן הנחה סמויה: כל קודקוד $v \in V$ מכיר את Δ.
2. נשווה עם השכנים. ישנם שני מקרים -

א. אף שכן לא בחר את הצבע שאנחנו בחרנו: במקרה זה, אנחנו הצלחנו. נחליט שזה הצבע שלנו.

ב. אחרת, קיים לפחות שכן אחד שבחר את הצבע שאנחנו בחרנו, במקרה זה אנחנו ננסה לבחור שוב את הצבע. (נחזור ל1).

נראה את האלגוריתם עבור קודקוד יחיד $u \in V$ ־

Color (Δ)**:**
while(true):
-pick random color from $[1, ..., 2\Delta] \to c$
- send c to neighbors
- recieve colors of neighbors
-if there is no neighbor with color c so return and update $C(u) = c$.

נשים לב שבבדיקה אנחנו בודקים את כל השכנים של הקודקוד, ולא רק את אלו ש"אקטיביים" כרגע. כלומר ־ בודקים גם את אלו שסיימנו לצבוע את הקודקוד שלהם.

0.6.1 נכונות האלגוריתם

מה הסיכוי שהבדיקה בשורה 5 תצליח? כלומר: שאין שכן שגם בחר את הצבע c.
לכל קודקוד יש דרגה $d \geq \Delta$ ולכן לא משנה איזה צבעים השכנים בחרו, תמיד יש לפחות $2\Delta - d$ צבעים פנויים. שורה 5 בהכרח מצליחה אם הצבע c שנבחר הוא אחד מהצבעים הפנויים. נסמן בx את מס' הצבעים הפנויים ברגע זה. בהכרח $x \geq 2\Delta - d$.

$$Pr[SuccessLine5] = \frac{x}{2\Delta} \geq \frac{2\Delta - d}{2\Delta} = 1 - \frac{d}{2\Delta} \geq_{\Delta \geq d} 1 - \frac{\Delta}{2\Delta} = \frac{1}{2}$$

כלומר, הסיכוי להצליח בשורה 5 הוא גדול שווה מ$\frac{1}{2}$.
כעת, נרצה לחסום את מס' הסיבובים שהאלגוריתם עולה עד לצביעה חוקית של כל הגרף.

נסמן בV_i את קבוצת הקודקודים שעדיין לא נצבעו אחרי i איטרציות של האלגוריתם. בהכרח לפי הגדרה $V_0 = V$. נראה כי

$$\forall u \in V : Pr[u \in V_i] \leq \frac{1}{2^i}$$

כיוון שהסיכוי שקודקוד יהיה בקבוצה, אזי בכל האיטרציות הקודמת היה כשלון בשורה 5. שכן אנו יודעים שכשלון בשורה 5 קטן שווה מסיכוי חצי.

$$E[|V_i|] = \sum_{u \in V} Pr[u \in V_i] \leq \frac{n}{2^i}$$

שכן $n = |V|$. לכן, אחרי $log(n) + 1$ איטרציות נקבל כי

$$E[|V_{logn+1}|] \leq \frac{n}{2^{logn+1}} = \frac{1}{2}$$

כלומר מס' הקודקודים בתוחלת לאחר $logn + 1$ איטרציות הוא חצי. נראה כי ישנה טעות נפוצה בשלב זה: להגיד מכאן נובע כי מס' האיטרציות עד שכל הקודקוד צבועים הוא לכל היותר "$logn + 1$". נראה כי זה שהתוחלת היא לכל היותר חצי (לא יתכן הרי חצי איבר), לא אומר שלאחר $logn + 1$ איטרציות הקבוצה תהיה ריקה (אין כאן לינאריות למעשה, זה שיש חצי איבר שנשאר זה לא גורר שלאחר $log(n) + 1$ איטרציות נסיים. התוחלת למעשה כאן אינה פונקציה הופכית ולכן הכיוון

ההפוך לא נכון). בהתחשב בתובנה הזו: כיצד נמשיך מכאן? נראה כי תמיד יתקיים לפי האלגוריתם וההסתברות שראינו קודם כי –

$$E[|V_i|] \leq \frac{1}{2}|V_{i-1}|$$

נזכר באי שוויון מרקוב. שאומר את הטענה הבאה: $Pr[X \geq t] \leq \frac{E[X]}{t}$. מכאן, נרצה לחשב מה הסיכוי שגודל של קבוצה יהיה גדול שווה מ$\frac{3}{4}|V_{i-1}|$.

$$Pr[|V_i| \geq \frac{3}{4}|V_{i-1}] \leq \frac{E[V_i]}{\frac{3}{4}|V_{i-1}|} \leq \frac{\frac{1}{2}|V_{i-1}|}{\frac{3}{4}|V_{i-1}|} = \frac{2}{3}$$

מכאן שסיכוי זה הוא לכל היותר $\frac{2}{3}$. ומכאן:

$$Pr[|V_i| < \frac{3}{4}|V_{i-1}] = 1 - Pr[|V_i| \geq \frac{3}{4}|V_{i-1}] \geq 1 - \frac{2}{3} = \frac{1}{3}$$

כלומר, הסיכוי שבאיטרציה הi הצלחנו לצבוע לפחות רבע מהקודקודים שלא היו צבועים קודם באיטרציה ה$i-1$ היא לפחות סיכוי של $\frac{1}{3}$. המספר שליש הוא קבוע, וזה מה שחשוב כאן. נסמן $p = \frac{1}{3}$.

נאמר שאיטרציה היא "טובה" אם היא הצליחה לצבוע לפחות רבע מהקודקודים שלא היו צבועים בתחילת האיטרציה: כלומר $|V_i| \leq \frac{3}{4}|V_{i-1}|$ (לכל היותר נשארו $\frac{3}{4}$ מכמה שהיו פעם קודמת). ורעה=לא טובה.

בהינתן שהיו k איטרציות טובות, נשארנו עם $(\frac{3}{4})^k \times n \geq$ מהקודקודים.

נרצה כי:

$$(\frac{3}{4})^k \times n < 1 \Longrightarrow n < (\frac{4}{3})^k \Longrightarrow log_{\frac{4}{3}}(n) < k$$

כלומר אם $k = log_{\frac{4}{3}}(n) + 1$ אזי הצלחנו לצבוע את כל הגרף. כלומר אם נקח k כנ"ל כמס' האיטרציות הטובות אנחנו סיימנו. נבחין: החישוב כאן הינו דטרמיניסטי לחלוטין, שכן אם היו k איטרציות טובות, נשארנו עם לכל היותר $(\frac{3}{4})^k n$ קודקודים, לכן קיבלנו חסם על k ובפרט את k.

קודם לכן ראינו כי $Pr[|V_i| < \frac{3}{4}|V_{i-1}] \geq \frac{1}{3}$, כלומר הסיכוי שתהיה איטרציה טובה הוא לפחות שליש. לכן, בתוחלת (התפלגות גאומטרית) עם $p \geq \frac{1}{3}$ מס' האיטרציות שיש ברצף עד שמקבלים איטרציה טובה ראשונה הוא $\frac{1}{p} \leq 3$.

ומכאן: מס' האיטרציות הטובות הינו $log_{\frac{4}{3}}(n)+1$, כפול 3 כמס' האיטרציות שקורות עד שמקבלים את האיטרציה הטובה הבאה (הסתברות היא $\frac{1}{3}$ וזהו משתנה גאומטרי). סה"כ נקבל כי

$$3log_{\frac{4}{3}}(n) + 3$$

הוא תוחלת מס' האיטרציות עד שאין קודקודים לא צבועים. ואכן, מס' האיטרציות שהאלגוריתם יעשה תהיה $O(logn)$.

כעת, נרצה **גם** לבחון את הנכונות במקרה הגרוע ביותר. ראינו כי

$$E[|V_i|] \leq \frac{n}{2^i}$$

נרצה לבדוק מה הסיכוי שלא סיימנו עבור $i = clogn$ עבור $c > 1$ קבוע.

$$Pr[|V_{clogn}| < 1] = 1 - Pr[|V_{clogn}| \geq 1]$$

$$Pr[|V_{clogn}| \geq 1]_{markov} \leq \frac{E[|V_{clogn}|]}{1} \leq \frac{n}{2^{clogn}} = \frac{1}{n^{c-1}}$$

ונקבל כי

$$Pr[|V_{clogn}| < 1] = 1 - Pr[|V_{clogn}| \geq 1] \geq 1 - \frac{1}{n^{c-1}}$$

כלומר, הסיכוי שלא סיימנו קטן פולינומית. ולכן הסיכוי להצלחה יחסית טוב.

הערה. מדובר באלגוריתם לאס וגאס כי הוא תמיד צודק ומצליח. בתחילה, חישבנו את זמן הריצה שלו בתוחלת.