אלגוריתמים 1 - הרצאה 12: MST מתקדם

27 בינואר 2026

גיא יער־און.

0.1 מבוא

הנחה לאורך כל ההרצאה: קיים יחס סדר מלא על הקשתות, נקרא לפונקציה זו $ID(e)$ לכל קשת $e \in E$. בהינתן שיש יחס סדר מלא על הקשתות ניתן לומר $e < e' \iff (w(e), Id(e) <_{Lex} (w(e'), Id(e')$ (מסתכלים ראשית על המשקל, אם הקשתות הן באותו משקל אזי ישנה דרך לשבור סימטריה לפי הId. מסקנה: כל הקשתות במשקלים (לפי יחס הסדר) שונים).

מסקנה: הרצה של קרוסקל לפי יחס סדר זה תניב תמיד עץ פורש מינימום יחיד (שכן תמיד ישנו אותו סדר למיון בשל שבירת הסימטריה).

נזכר כי ראינו שתי תכונות עיקריות:

1. קשת קלה בחתך $\Longleftarrow$ קיים MST שמכיל את e.
2. קשת כבדה במעגל $\Longleftarrow$ קיים MST שלא מכיל את e.

בהקשר של יחס סדר המלא: קשת e שקלה בחתך (עם הid הכי קטן) תהיה בכל MST
קשת e שביחס סדר מגיעה הכי מאוחר (עם id הכי גדול) לא תהיה באף MST.

0.2 $Boruvka$

מדובר באלגוריתם הראשון שהומצא לMST.

נסביר את הרעיון ללא פסאודו קוד:

מתחילים מאוסף של קודקודים של הגרף, ישנן קשתות בגרף אך לא מציירים אותן עדיין - כל קודקוד בוחר את הקשת הקטנה ביותר שלו לפי יחס הסדר.

נבחין כי ישנם שני מקרים:

אם עבור קודקוד v הקשת הכי קטנה היא (v,u) והקשת הכי קטנה עבור u היא גם (v,u) אזי מעולה. אך זה לא בהכרח - יתכן שלu יש קשת טובה יותר טובה.

נבחין: בהכרח לא יווצר מעגל! בשלילה שיש מעגל אזי יש קודקוד ש2 קשתות חלות בו, ונקבל סתירה ליחס הסדר במעגל (יהיה קודקוד שבחר בקשת הלא טובה ביותר שלו).

טענה: בעקבות יחס הסדר, הקשתות שנבחרו בסיום התהליך לא יצרו מעגל.

נוסיף את כל הקשתות לMST, נסמנו E_T.

בשלב הבא, נכווץ את הקשתות שנבחרו (ננסה לדמיין את התהליך ככה שבתחילה כל קודקוד הוא רכיב קשירות בפני עצמו. אם קודקוד 1 וקודקוד 2 חוברו בניהם כעת הם נחשבים לקודקוד אחד בלבד 12 - קודקוד גדול!).

נבחין כי אם קודם היו $|V|$ קודקודים, לאחר הכיווץ יהיה לכל היותר $\frac{|V|}{2}$ קודקודים (המקרה הכי טוב שכל שניים בחרו אותה קשת)

נמשיך ברקורסיה עד למקרה הבסיס (קודקוד אחד) - יהיה לנו סה"כ $O(log|V|)$ איטרציות.

מעט יותר פורמלית:

בכל איטרציה:

א. כל הקודקודים בגרף הם מבודדים (או מגה הקודקודים שיצרנו)

ב. לכל קודקוד $v \in V$ מצא את הקשת e_v המינימלית המחוברת אליו.

ג. הוסף את כל הקשתות e_v שמצאת אל MST

ד. כיווץ: אגד את כל הקודקודים שחוברו ע"י הקשתות הללו ל"קודקודים גדולים" ומחק לולאות עצמיות (קשתות שחיברו בין קודקודים שכעת באותו קודקוד) - הקשתות שישארו לך בגרף הם רק קשתות שחיברו בין גושים של קודקודים גדולים.

נבחין כי עלינו למצוא דרך להפטר מהלולאות העצמיות - בכל כיווץ נפעיל DFS למצוא את הרכיבים הקשירים ואם קשת ששני קצוותיה היו באותו רכיב מדובר כעת בלולאה העצמית שנפטר ממנה. זה לא יפגע בזמן הריצה אסימפטוטית. (הערה: המימוש הטכני תלוי לפרשנות וישנם דרכים רבות. למשל: בעת שהוא מאחד את הקודקודים הוא נותן להם ID חדש לכל אחד מהם, ובסיבוב הבא שיריץ DFS ויראה קשת ששני הקודקודים בה מרכיב עם אותו ID הוא יתעלם מהקשת והוא יכריז עליה כלולאה עצמית).

זמן הריצה:

נפרק את הניתוח לשניים:

1. כמה עולה סבב?

* בכל סבב כל קודקוד עובר על כל קשתותיו ומוצא את הקשת המינימלית. לכן $O(\sum_{v\in V} deg(v)) = O(|E|)$

*בשביל לדעת איזה קודקוד התאחד עם מי, מריצים $DFS\backslash BFS$ על הקשתות שבחרנו + עוברים על הקשתות ומוחקים את אלו שהפכו ללולאות עצמיות. סה"כ $O(|V|+|E|) = O(|E|)$

לכן סבב עלותו $O(|E|)$

2. כמה סבבים ישנם?

נסמן בV_i את מס' הקודקודים בשלב הi. נבחין כי בהכרח $|V_i| \leq \frac{1}{2}|V_{i-1}|$ (במקרה הגרוע ביותר: כל זוג קודקודים מתאחדים לקודקוד אחד) ולכן מס' האיטרציות הינו $O(log|V|)$

לכן סה"כ זמן הריצה הוא $O(|E|log|V|)$.

0.3 שיפור Yao

יאו ישב ואמר: בורובקה עשה רעיון יפה - אך למה שלכל קודקוד נסרוק את כל הקשתות שיוצאות ממנו? אם אנחנו יודעים כי קשת היא מאוד יקרה - אולי לא כדאי להסתכל עליה בסיבובים הראשונים.

יאו הסתכל על קודקוד $u \in V$ ועל כל הקשתות שיצאו ממנו (ישנן $deg(u)$ קשתות כאלו) וחילק את קשתות u לk קבוצות $(Buckets)$: נסמן $E_u^{(1)}, ..., E_u^{(k)}$ ועבור כל קבוצה שכזו מתקיים כי גודלה הוא $\left\lceil \frac{deg(u)}{k} \right\rceil$.

יחס הסדר צריך לקיים כי כל מי שבקבוצה הראשונה קטן ממי שבקבוצה השנייה, ומי שבשנייה קטן מהשלישית וכו'.

כיצד יוצרים את החלוקה הזו? נניח כי k חזקה של 2, ונשתמש באלגוריתם $Select$ למציאת החציון. מכאן נקבל שתי קבוצות (גדולים וקטנים מהחציון), ונמשיך ברקורסיה על כל צד (של אלו שגדולים מהחציון ואלו שקטנים ממנו). כך נמשיך ברקורסיה - עד שנקבל k קבוצות בדיוק. למעשה, כיוון שבכל שלב אנחנו מחלקים את הקבוצות ל2, מס' הפעמים שנצטרך לבצע את הרקורסיה הוא $log(k)$. כמה עולה כל הרצה? אלגוריתם סלקט עולה זמן לינארי. כאן הלינאריות הינה $O(deg(u))$

- רק הקשתות של אותו קודקוד. לכן סה"כ זמן הריצה $O(deg(u) \times log(k)$

כעת, קודקוד $u \in V$ צריך לחפש את הקשת הקלה ביותר שיוצאת ממנו. הוא לא צריך (בזכות עבודת העזר המקדימה) לעבור על כל השכנים שלו. אלא: רק על $E_u^{(1)}$. לכל קשת הוא יבדוק: האם הקשת מחברת אותי לקודקוד שנמצא איתי ברכיב?

אם כן - מדובר בלולאה עצמית. האלגוריתם יזרוק את הקשת הזו וימשיך לקשת הבאה באותה חבילה. נבחין: הוא יעיף אותה לתמיד.

אם לא - מצאנו את הקשת הרלוונטית. זו הקשת הכי זולה שיוצאת מקודקוד u. הוא יבחור אותה לסיבוב הנוכחי של בורבוקה ולא יפתח את שאר החבילות. הוא יעבור לחבילה הבאה - רק כאשר יסיים עם הקשתות בחבילה הנוכחית. הנכונות מגיעה מכך שבהכרח הקשתות בחבילות הבאות - גדולות מהקשתות בחבילה שלו.

קודקוד יגדיר עבור חבילה שהיא כבר לא רלוונטית - אם ורק אם כל הקשתות שנמצאות בה גולו כלולאות עצמיות (קשתות שמחוברות לקודקוד באותו רכיב).

ניתוח זמן הריצה:

העלות מורכבת מ3 חלקים:

1. **עיבוד מקדים:** לכל קודקוד $u \in V$ חישוב ה$Buckets$ שלו: $O(\sum_{v \in V} deg(v) \times log(k)) = O(|E|logk)$
2. **סריקה של קשתות זבל:** נבחין כי קשת זבל (שהתגלתה כקשת עצמית) נסרקת כזבל בדיוק פעם אחת - כשמגלים שהיא זבל מסירים אותה. ולכן $O(1)$ פר קשת כלומר $O(|E|)$
3. **חיפוש קשת בכל סבב:** בכל אחד מהסבבים של בורבוקה כל קודקוד סורק חבילה אחת (או חלק ממנה). נבחין כי הקודקוד יודע באיזו קבוצה עליו לחפש כי הוא העיף כבר (חישבנו את החישוב סה"כ עבור האלגוריתם) את הקשתות שהפכו לזבל. גודל כל חבילה הינו $\left\lceil \frac{deg(v)}{k} \right\rceil$ ולכן עבודה עבור כל הקודקודים בסבב היא בעלות:

$$O(\sum_{v \in V} \left\lceil \frac{deg(v)}{k} \right\rceil) \leq \frac{|E|}{k} + |V| = O(\frac{|E|}{k} + |V|)$$

סה"כ ישנם $log(|V|)$ סבבים אצל בורבקה ולכן נקבל שזמן הריצה הינו:

$$log(|V|) \times (\frac{|E|}{k} + |V|) + |E| + |E|logk$$

נגדיר: $k = \lceil log(|V|) \rceil$ ונקבל זמן ריצה:

$$O(|V|log|V| + |E|log(log(|V|))$$

המטרה: להפטר מ$|V|log|V|$.

נחלק את האלגוריתם ל2:

1. נריץ את האלגוריתם הרגיל של בורבוקה (זה שעלותו $O(|E|)$ למציאת הקשתות סה"כ בסבב) למשך $log(log|V|)$ סבבים בלבד. עלות חלק זה: $O(|E|loglog(|V|)$

הבחנה: כיוון שבכל סיבוב מספר הקודקודים קטן פי 2 (לפחות) נקבל כי אם נסמן בV' את גודל קבוצת הקודקודים לאחר ההרצות של בורבוקה אזי היא:

$$|V'| = \frac{|V|}{2^{log(log(|V|))}} = \frac{|V|}{log(|V|)}$$

2. כעת, נריץ את האלגוריתם של Yao על הגרף שנותר עם V'. נבחין כי $|V'| \leq |V|$ והאלגוריתם של יאו יעלה לנו:

$$|V'|log|V'| + |E|log(log(|V'|))$$

$$\leq_* \frac{|V|}{log|V|} \times log(|V|) + |E|log(log(|V|)) = O(|V|) + O(|E|log(log(|V|)) = O(|E|log(log(|V|))$$

הסבר ל$*$: הפעלנו את $|V'| = \frac{|V|}{log|V|}$ רק על החלק $|V'|$ הראשון והשתמשנו בכך שבהכרח $|V'| \geq |V|$ בשאר המיקומים של $|V'|$, ובחלק האחרון השתמשנו בכך ש$O(|V|) = O(|E|)$ כי הגרף קשיר.

0.4 האלגוריתם של (KKT) $Karger - Klein - Tarjan$

כעת נראה אלגוריתם שרץ בתוחלת $O(|E|)$ זמן למציאת MST.

נסמן $F(Forest)$ יער על V (חלק מהגרף $G = (V, E)$: מדובר ביער כלשהו של הגרף).

הגדרה: יהיו u, v באותו רכיב קשירות של F. נסמן ב$P_F(u, v)$ את המסלול הפשוט היחיד בין u ו v בF. (קיים מסלול יחיד בעץ)

הגדרה: $max_F(u, v) = max\{e | e \in P_f(u, v)\}$ (משקל הקשת הכבדה ביותר על המסלול הפשוט מu לv. כבדה ביותר ביחס ליחס סדר!)

הערה חשובה להמשך ומוטיבציה (אינטואיציה, זה אני הוספתי לא היה בהרצאה): אלגוריתם KKT מנסה להיות עצלן - במקום לעבוד קשה ולמצוא את הMST האמיתי של E הקשתות הוא דוגם חלק קטן מאוד מהקשתות. הוא ימצא את הMST שלהן (היער F) והוא ישתמש בו בכדי לסנן את שאר הקשתות בגרף המקורי. זה יהיה יער שנוצר מדגימה אקראית. למען האינטואיציה עוד: נניח כי יש לנו גרף ענק, נזרוק לכל קשת מטבע: אם יצא עץ - היא תכנס לקבוצה בשם E_s, אחרת לא תכנס. כעת נמצא את הMST של הקבוצה הקטנה הזו, התוצאה היא יער שלו נקרא F.

הגדרה: קשת $e = (x, y)$ נקראת F כבדה אם:

1. x, y באותו רכיב לפי F (למה? אם הם לא מחוברים בF, אין לנו מסלול להשוות אליו ואז לא ניתן לומר בהכרח שהיא מיותרת).
2. $e > max_F(x, y)$

אחרת: נאמר כי e היא F קלה.

למה: e היא F כבדה $\Longleftarrow$ e לא בMST. (זה חלק גדול מלב האלגוריתם. האלגוריתם אומר שלמרות שהשווינו אותה רק ליער חלקי F העובדה שהיא סגרה מעגל שבו היא כבדה מספיק בכדי לומר: את לא בMST בהכרח כי הוא יחיד)

קופסה שחורה: בהינתן F, V, E קיים אלגוריתם שמוצא את כל הקשתות הF כבדות בזמן לינארי

.$O(|E|)$

טענה (**משפט דגימה**): בהינתן $G=(V,E)$, נייצר גרף אקראי $H=(V,E_S)$ כך שכל קשת $e\in E$ נבחרת בהסתברות $0<p\le 1$. יהי F הMSF (יער פורש מינימום של H), אזי $\mathbb{E}[X]\le\frac{|V|-1}{p}$ (באשר X מ"מ שסופר את מס' הקשתות בE שהן F קלות).

הסבר רעיוני. כלומר, אם דגמנו יער כאשר קשת נכנסה אליו בהסתברות p, ונבנה מהן MSF, מס' הקשתות הF קלות (אלו שלא הצלחנו לזרוק) הוא לכל היותר $O(\frac{|V|}{p})$.

הוכחה:

נניח לצורך ההוכחה (זו לא באמת הדרך שבה מצאנו את F!) כי השתמשנו בקרוסקל על H על מנת למצוא את F. (זה לא אומר שהאלגוריתם שהרצנו הוא בהכרח של קרוסקל! אך כאן זה מותר להניח זאת שכן ישנו MSF יחיד ולכן כל האלגוריתמים יחזירו את אותו מבנה בהכרח).

כלומר, קרוסקל מורץ על הגרף הדגום H.

נתבונן על הרגע שבו קרוסקל הוסיף קשת $e=(x,y)$ לF.

בזמן זה, נסמן בA את הרכיב שמכיל את x ובB את הרכיב שמכיל את y.

כעת אנחנו יכולים להניח כי הקשת e היא הקלה ביותר בחתך בH שמחברת את A וB כלומר היא הקשת הקלה ביותר בחתך (A,B). נתבונן על כל הקשתות בגרף המקורי! שחוצות את החתך בין A לB. חלק מהן נדגמו לH וחלק לא.

הקשת שקרוסקל בחר, היא הקשת הראשונה e מתוך הרשימה הנ"ל שאכן נדגמה לH.

כמה קשתות היינו צריכים לדגום עד שנגיע לקשת הראשונה הזו? מדובר ב$Geo(P)$ - בודקים עד להצלחה. תוחלת מספר הקשתות שנבדקו עד ההצלחה הראשונה היא $\frac{1}{p}$.

כל הקשתות שהיו לפני e ברשימה (אלו שלא נדגמו) הן בהכרח F קלות. מדוע? כאשר יער P יושלם, לא יהיה בינן לבין e מסלול שכולו קשתות יותר (הרי הן בהכרח לא נכנסו! היא הראשונה שנכנסה!)

הבחנה: כל הקשתות שמגיעות ביחס הסדר לאחר e הן בהכרח F כבדות.

הסבר: ברגע שקרוסקל הכניס את e לF, הוא חיבר את A וB. כל קשת אחרת e' בחתך (A,B) שגדולה מe: $e'>e$ סוגרת בהכרח מעגל עם הקשתות שכבר בתוך A וB (הרי קשת שחוצה את החתך (a,b) היא קשת כך ש$a\in A, b\in B$ וכן אם נסמן את הקשת הקלה בחתך $e=(x,y)$ אזי בA ישנו מסלול P_{ax} ובB ישנו מסלול P_{yb} ולכן $P_{ax}\circ(x,y)\circ P_{yb}\circ(b,a)$ הוא מעגל: שבו הקשת e' היא בהכרח הקשת הכבדה ביותר - ולכן בהכרח $e'=(a,b)$ כך ש$e'>e$ היא קשת שכבדה ביותר במעגל ולא נמצאת באף MSF. מדובר בקשת F כבדה!). ניתוח זה נכון לכל $e'>e$.

כעת, כל קשת שנכנסה לF (ישנן לכל היותר $|V|-1$ כאלו) ייצגה בתוחלת $\frac{1}{p}$ קשתות לכל היותר שהיו רלוונטיות להיות פוטנציאליות קלות בחתך שלה. קשתות שלא היו בחתך של אף קשת הן בהכרח כבדות כפי שאמרנו. לכן e מגדירה $\frac{1}{p}$ קשתות קלות לכל היותר. לכן: מס' הקשתות הקלות בתוחלת לכל היותר חסום ב:

$$\mathbb{E}[X]\le\frac{1}{p}\times(|V|-1)=\frac{|V|-1}{p}$$

0.5 פסאודו של KKJ

:$ALG(V,E)$

א. אם $E=\emptyset$ החזר $\emptyset$.

ב. הרץ 2 סבבים של $Boruvka$ (נשארו $\frac{|V|}{4}$ קודקודים לכל היותר) לקבלת יער קשתות B.

ג. כווץ קשתות B בG. נסמן את הגרף המכווץ $G'=(V',E')$

ד. כל קשת מE' מצטרפת לE_S בסיכוי $p=\frac{1}{2}$ (דגימת קבוצת הקשתות שנשארו לאחר הכיווצים)

ה. $F=ALG(V',E_S)$ (מדובר ביער פורש מינימום על הקשתות שדגמנו)

ו. מצא את כל הקשתות הF כבדות בG' (באמצעות קופסה שחורה) ומחק אותן מE'.

ז. נסמן בL את הקשתות שנשארו בG' (נותרו בתוחלת $2|V'| \approx \frac{|V'|-1}{\frac{1}{2}}$)

הערה. למעשה בדוגמה הקודמת כל מי שנכנסות אל L הן אלו שמתחת לe ואלו שלא הן בהכרח הF כבדות שיועפו ולא יכנסו. נבחין כי בהכרח קשתות L מכילות MSF (ועוד קשתות!)

ח. $F' = ALG(V', L)$

ט. החזר $B \cup F'$

האלגוריתם מחזיר לבסוף MST כיוון שהגרף במקור הוא קשיר מהנחתנו והיער הפורש המינימלי שלו הוא עץ.

הסבר:

א. בבירור: אם אין קשתות, אין מה לחבר בגרף. תנאי עצירה.

ב+ג: עושים מעין דיאטה" לקודקודים. אנחנו מריצים שתי הרצות של בורבוקה. מדוע דווקא 2? זה לא קריטי דווקא 2 - יכולנו גם 3 וכו' אך חשוב לא הרצה אחת. זה יהיה חשוב לניתוח זמן הריצה בתוחלת בהמשך. נסמן בB את הקשתות שבורבוקה בוחר: לפי תכונת החתך - הן בהכרח בMST (ונכונותו של בורבוקה, אלו בהכרח קשתות שנמצאות בMST). נבחין כי לאחר שתי הרצות נקבל שבגרף שקבוצת קודקודיו (המכווצים) היא לכל היותר בגודל $\frac{|V|}{4}$. כעת: עוברים לעבוד על גרף הרבה יותר קטן מבחינת קודקודים.

ד: כעת מבצעים דגימה: S. כל קשת מE' (אלו שנותרו בגרף המכווץ) תצטרף לגרף הדגימה E_S בהסתברות $\frac{1}{2}$.

ה: מבצעים רקורסיה: מבקשים מהאלגוריתם למצוא את הMSF של הקבוצה הקטנה הזו. שוב ושוב - עד שנגיע לכך שאין קשתות.

בסיום שלב ה', בידינו MSF (יער פורש מינימלי) של הקשתות שדגמנו. F הוא לא הMST של הגרף המקורי! הוא הMST של חלק מהקשתות (בתוחלת, חצי מהקשתות) אותם בחרנו באקראי.

ו: כעת, יש בידינו MSF שחזר, אנחנו יודעים לפי מה שטענו קודם כי הקשתות הכבדות לא יהיו בו. באמצעות הקופסה השחורה - אנחנו נמצא את כל הקשתות ופשוט נעיף אותן מE'. למעשה: אנחנו יודעים לזהות על הרבה קשתות כרגע שאינן בMST הסופי.

בשלב זה - הקשתות מG' שלא נבחרו (בתוחלת חצי מהן לא נבחרו) חוזרות לתמונה. קודם לכן הרצנו יער בלעדיהן, כאן הן חוזרות לתמונה.

ז: כל הקשתות שנשארו לאחר המחיקה, יסומנו בL. אלו שלא הצלחנו לזרוק - ישנן בתוחלת $O(|V'|)$ כאלו, מדובר במספר דליל מאוד של קשתות! לינארי במספר הקודקודים.

ח: בסוף, כשיש לנו גרף רזה מאוד, מריצים עליו את האלגוריתם עם קבוצת הקשתות שלא הצלחנו למחוק L וV'.

ט: התוצאה מהרקורסיה תהיה הMSF האמיתי של הגרף המכווץ.

סה"כ: B אלו הקשתות שאנחנו יודעים שבהכרח בMST מנכונות בבורסקה, וF' זו תוצאת הMSF על הגרף המכווץ (לאחר הרצת פעמיים של בבורסקה) ולכן סה"כ העץ שלנו מורכב מ$B \cup F'$.

0.6 זמן הריצה

על מנת לנתח את זמן הריצה נסמן $|V| = n, |E| = m$:

$$T(n,m) = O(m+n) + T(n_1, m_1) + T(n_2, m_2)$$

נבחין כי כל הפעולות הן בזמן לינארי פרט לרקורסיות: פרט לזוג הרקורסיות כל העבודה שמתבצעת לינארית ותסומן ע"י $O(n,m)$.

$T(n_1, m_1)$ הוא הקריאה הראשונה לרקורסיה ו$T(n_2, m_2)$ הוא הקריאה השנייה לרקורסיה.

נרצה לחשב את $\mathbb{E}[T(n,m)]$

$$\mathbb{E}[T(n,m)] = O(m+n) + \mathbb{E}[T(n_1, m_1)] + \mathbb{E}[T(n_2, m_2)]$$

נוכיח באינדוקציה כי $\mathbb{E}[T(n,m)] \leq 2 \times a(\overline{m}+2\overline{n})$ באשר $\overline{x}=\mathbb{E}[X]$. וכן $a>1$ קבוע (מספיק גדול כדי לחסום את $m+n$ כלומר $O(m+n) \leq a(m+n)$.)

משפט 1. מתקיים $\mathbb{E}[T(n,m)] \leq 2a(\overline{m}+2\overline{n})$

הוכחה: נוכיח את הטענה באינדוקציה על $n+m$.

בסיס: טריוואלי.

צעד: נכונות לכל $n'+m' < n+m$. נוכיח עבור $n+m$.

$$\mathbb{E}[T(n,m)] = \mathbb{E}[a(m+n)] + \mathbb{E}[T(n_1,m_1)] + \mathbb{E}[T(n_2,m_2)]$$

$$\leq_* a\overline{m} + a\overline{n} + 2a(\overline{m_1}+2\overline{n_1}) + 2a(\overline{m_2}+2\overline{n_2}) =$$

* הוא הנחת האינדוקציה.
נבחין כי $\overline{n_1},\overline{n_2} \leq \frac{\overline{n}}{4}$ (באמצעות שני סבבי בורבוקה)
וכן $\overline{m_1} = \mathbb{E}[|E_S|] = \frac{|E'|}{2} \leq \frac{|E|}{2} = \frac{\overline{m}}{2}$ (קשתות נכנסות בהסתברות חצי).
וכן $\overline{m_2} \leq 2\overline{n_2} \leq 2 \times \frac{\overline{n}}{4} = \frac{\overline{n}}{2}$ (לפי משפט הדגימה).
כעת נציב:

$$\leq a\overline{m} + a\overline{n} + 2a(\frac{\overline{m}}{2}+\frac{\overline{n}}{2}) + 2a(\frac{\overline{n}}{2}+\frac{\overline{n}}{2}) =$$

$$= 2a(\overline{m}+2\overline{n})$$

כנדרש.

מסקנה: זמן הריצה של האלגוריתם $O(n+m)=O(m)=O(|E|)$ בתוחלת.

בהצלחה לכולנו במבחן!