סיכומים למבחן בקורס אלגוריתמים

סמסטר א' 2008-9 (פרופ' מיכה שריר)

גרפים / חזרה כללית:

מושגים:

- **גרף**: גרף $G=(V,E)$, V קבוצת קודקודים, $E\subset V\times V$ קבוצת קשתות. מכוון: הקשתות הן זוגות סדורים, לא מכוון: הקשתות הן קבוצה בת שני איברים (הסדר הפנימי לא חשוב).
- **לולאה**: (a,a) - קשת מקודקוד לעצמו.
- **קשתות מקבילות ואנטי מקבילות**: קשתות בין אותם שני קודקודים (גרף לא מכוון; בגרף מכוון: כיוונים זהים). קשתות אנטי מקבילות: בגרף מכוון בלבד, קשתות בין אותם שני קודקודים אך בכיוונים הפוכים.
- **גרף פשוט**: גרף ללא קשתות מקבילות, אנטי מקבילות ולולאות.
- **שכנות (צמתים)**: תהי קשת (a,b), אזי a,b הן נקודות קצה של הקשת, ו-a ו-b שכנים אחד של השני.
- **סמיכות (קשתות)**: זוג קשתות יקראו סמוכות אם יש להן נקודת קצה משותפת.
- **דרגת צומת**: מספר הקשתות היוצאות ממנו והנכנסות אליו. הגדרת דרגת יציאה ודרגת כניסה בהתאם.
- **קודקוד מבודד**: קודקוד שדרגתו 0. אם רק דרגת הכניסה שלו 0, יקרא **מקור**, ואם רק דרגת היציאה שלו 0, יקרא **בור**.
- **תת-גרף**: של $G=(V,E)$ הוא גרף $G'=(V',E')$ כך ש-$V'\subseteq V$, $E'\subseteq E$ ו-$E'\subseteq V'\times V'$.
- **תת-גרף מושרה**: תת-גרף הכולל את <u>כל</u> הקשתות ב-E המחברות את הצמתים ב-V'.
- **מסילה/מסלול**: סדרת צמתים $v_1,\dots,v_k$ כך שלכל $1\le i\le k-1$: $(v_i,v_{i+1})\in E$. אורך המסלול = מספר הקשתות בו.
- **מעגל**: מסלול כך ש-$v_1=v_k$.
- **מסלול/מעגל פשוט**: אין חזרה על צמתים.
- **קליק (*Clique*, לא מכוון)**: תת קבוצה של צמתים בה כל צומת מחובר לכל צומת אחר.
- **גרף שלם**: גרף בו V מהווה קליק, כלומר כל צומת בגרף מחובר בקשת לא מכוונת לכל צומת אחר בגרף.
- **קבוצה ב"ת (*IS*)**: ההיפך מקליק, תת קבוצה של צמתים בה אף זוג צמתים אינו מחובר בקשת.
- **גרף דו-צדדי (לא מכוון)**: V הוא איחוד זר של 2 תתי קבוצות V_1,V_2 ו-$E\subseteq V_1\times V_2$. כלומר, כל קשת מחבר בין צומת ב-V_1 לצומת ב-V_2.
- **נגישות**: צומת u יהיה נגיש מצומת v אם קיים מסלול מ-v ל-u (כל צומת נגיש מעצמו).
- **גרף קשיר (לא מכוון)**: גרף בו כל צומת נגיש מכל צומת אחר בגרף.
- **רכיב קשירות**: אם גרף אינו קשיר, רכיבי קשירות הם תתי-קבוצות מקסימליות של צמתים מהגרף כך שהגרף המושרה של כל תת-קבוצה כזו הוא גרף קשיר. כל גרף מתפרק לרכיבי קשירות <u>זרים</u> בצמתים ובקשתות.
- **גרף קשיר בחוזקה**: כל צומת נגיש מכל צומת אר בגרף במסלול <u>מכוון</u>.
- **רכיב קשירות חזקה (רק"ח)**: אם גרף מכוון אינו קשיר חזקה, ניתן לפרקו לרכיבי קשירות חזקה שהם תתי-קבוצות מקסימליות של צמתים שתת הגרף המושרה של כל תת-קבוצה הוא גרף קשיר בחוזקה. בניגוד למקרה הלא מכוון, כאן <u>יתכנו</u> קשתות בין רק"חים.
- DAG **- גרף אציקלי מכוון**: גרף מכוון חסר מעגלים.
- **עץ**: DAG שהוא גם קשיר. אם אינו קשיר, אך כן חסר מעגלים, יהיה **יער** (כל רכיב קשירות שלו הוא עץ). תכונות עץ:
  - קשיר.
  - חסר מעגלים.
  - בעל $|V|-1$ קשתות <u>בדיוק</u>.

  } כל שתיים מהתכונות האלה גוררות שזהו עץ
  - בין כל שני צמתים קיים מסלול אחד ויחיד.
- **עץ מכוון**: עץ בו צומת כלשהו מוגדר כשורש, וכל הקשתות מכוונות אליו / ממנו.

ייצוג של גרפים:

1. **רשימות שכנות**: לכל צומת יש מצביע לרשימת הצמתים השכנים לו. יתרון: זול בזיכרון, $O(|V|+|E|)$; חסרון: יקר להגיע לכל שכן.
2. **מטריצת שכנויות**: מטריצה $|V|\times|V|$ בה $A_{ij}=\begin{cases}1,(i,j)\in E\\0,o/w\end{cases}$. יתרון: גישה מהירה; חסרון: בזבוז מקום (עבור גרפים מועטי קשתות).

BFS: Breadth First Search : חיפוש לרוחב:

עבור גרף נתון $G=(V,E)$ (מכוון או לא, נתייחס למכוון) וצומת התחלה s, האלגוריתם מבקר בכל צמתי הגרף <u>הנגישים מ-s</u> בלבד. הבדיקה נעשית על צמתים לפי סדר מרחקם מ-s: לכל שכבת מרחק תבוצע הבדיקה על כל הצמתים באותה שכבה לפני מעבר לשכבה הבאה. נחשב לכל צמתי הגרף את המק״ב (מסלול קצר ביותר) מ-s אליהם. לכל $u\in V$ נחזיק:

- $d[u]$: אורך המסלול הקצר ביותר מ-s אל u שנתגלה עד כה.
- $\pi[u]$: מצביע לצומת הקודם באותו מק״ב שנתגלה עד כה.

<u>**אופן פעולת האלגוריתם**</u>:

משתמשים בתור המחזיק תחילה את s עם שדות מאותחלים: $d[s]=0, \pi[s]=null$. עבור כל צומת u שנוציא מהתור, החל ב-s, נבצע:

- לכל שכן v הנגיש מ-u נעדכן את שדותיו: $d[v]=d[u]+1$, $\pi[v]=u$.
- לאחר עדכון כל השכנים נצבע את u באפור, אינדיקציה לכך שעברנו עליו.
- נוציא את הצומת הבא מהתור ונבצע את אותן פעולות עד סיום. אם צבעו אפור, נדלג עליו (כבר בוצעו עליו הפעולות).

<u>**סיבוכיות**</u>: $O(|V|+|E|)$ (לינארית).

<u>**סימון למק״ב**</u>: $\delta(s,u)$ = אורך המסלול הקצר ביותר מ-s אל u, או ∞ אם לא קיים מסלול מ-s אל u.

<u>**תכונות וטענות**</u>:

1. כל תת מסלול של מק״ב הוא מק״ב.
2. אם $E \ni (u,v)$ אזי $\delta(s,v)\le\delta(s,u)+1$.
3. אם (u,v) הקשת האחרונה במק״ב מ-s אל u, אז $\delta(s,v)=\delta(s,u)+1$.
4. אם u נגיש מ-s, אזי האלגוריתם מבקר בצומת u.
5. בכל רגע, בתור Q נמצאים צמתים עם לכל היותר 2 ערכי d עוקבים בסדר d עולה.

<u>**משפט**</u>: לכל צומת u מתקיים: $d[u]=\delta(s,u)$ (טענת עזר: במהלך ריצת האלגי מתקיים לכל u: $d[u]\ge\delta(s,u)$).

<u>**שימושים**</u>:

- שאלת הנגישות מ-s.
- מציאת מק״בים מכל צומת נגיש מ-s. המק״ב יהיה רשימת הצמתים ההפוכה למצביעי ה-π מ-u.

<u>**הערות**</u>:

- האלגי מסדר את הצמתים הנגישים מ-s בשכבות לפי ערך ה-d שלהן. למשל s יהיה הצומת היחיד בשכבה 0.
- בגרף מכוון: כל קשת שאינה בשימוש ה-BFS מחברת כל צומת בשכבה j לצמתים בשכבה $\le j+1$. בגרף לא מכוון: כל קשת מחברת צמתים באותה שכבה או בשכבות סמוכות.
- מצביעי ה-π מהווים עץ מכוון שקשתותיו מצביעות לכיוון השורש, ושורשו הוא s. זהו עץ המק״בים.

<u>**טענה**</u>:

<u>גרף לא מכוון הוא דו-צדדי $\Leftrightarrow$ לא קיים בו מעגל באורך אי זוגי.</u>

$\Leftarrow$: אם גרף הוא דו״צ, כל צומת מעבירה אותנו מ-V_1 אל V_2 וחזרה, ולכן כדי לחזור לאותו צומת צריך מספר זוגי של קשתות במעגל.

$\Rightarrow$: מריצים מצומת s כלשהו BFS. מסתכלים על חלוקת הצמתים לפי השכבות המתקבלות מה-BFS. נטען שלא קיימות קשתות המחברות שני צמתים באותה שכבה j, אחרת היינו מקבלים מעגל שאורכו $2j+1$ (אי זוגי, בניגוד להנחה). לפיכך, כל קשת מחברת לצומת בשכבות הסמוכות (מההערות לעיל), ולכן נוכל לחלק את צמתי הגרף לשתי קבוצות לפי השכבות הזוגיות והאי זוגיות. לכן הגרף הוא דו״צ.

<u>**תרגילים לדוגמא (מהתרגול)**</u>:

1. עבור $G=(V,E)$ (מכוון או לא), צור את גרף המק״בים מ-$s\in V$.

פתרון: מריצים BFS מ-s, ונגדיר את הגרף החדש G' באופן הבא: $V'=\{v\in V|\ d(v)<\infty\}$, $E'=\{(u,v)\in E|\ d(v)=d(u)+1\}$. אם הגרף לא מכוון, אז $E'=\{(u,v)\in E|\ d(v)=d(u)+1$ או $d(u)=d(v)+1\}$.

2. עבור $G=(V,E)$, צור את גרף המק״בים מ-$s\in V$ ל-$t\in V$.

פתרון: ניצור את גרף המק״בים מ-s שנסמנו G'. אם t לא מופיע בו אזי אין מסלול מ-s אל t ונחזיר את הגרף הריק. אחרת נהפוך את כיווני הקשתות ב-G' ונמצא גרף מק״בים מ-t מתוך G', נסמנו G''. נהפוך חזרה את קשתות G'' ונחזיר אותו – הוא גרף המק״בים מ-s אל t.

<u>מסלול ומעגל אויילר (מהתרגול)</u>:

<u>הגדרה</u>:

- **מעגל אויילר**: מעגל (לא בהכרח פשוט) שעובר על כל קשת בדיוק פעם אחת.
- **מסלול אויילר**: אותו דבר רק מסלול.

<u>טענות</u>:

- גרף <u>לא מכוון</u> וקשיר מכיל מעגל אויילר $\Leftrightarrow$ דרגות כל הצמתים בו זוגיות. כמו כן ניתן למצוא מעגל אויילר ב-$O(|E|)$.

רעיון ההוכחה: אם גרף מכיל מעגל אויילר, כל פעם שנכנסים לצומת u גם יוצאים ממנו, לכן כל מעבר בצמתים הוא במספר זוגי של קשתות. צומת ההתחלה: הקשת הראשונה תורמת 1, האחרונה 1, וכל מעבר 2, לכן סה"כ זוגי. אם דרגות כל הצמתים זוגיות, נבנה מעגל באופן הבא: מתחילים מצומת u כלשהו. ממשיכים במסלול ממנו עד שחוזרים אליו, כאשר כל קשת שעוברים בדרך מוציאים מ-E. אם נותרו קשתות ב-E, נחזור על שלבים אלו החל מצומת u הנמצא על המעגל שבנינו עד כה ונותרו לו קשתות ב-E. נמשיך עד ש-$E=\emptyset$.

- גרף <u>לא מכוון</u> וקשיר מכיל מסלול אויילר $\Leftrightarrow$ בגרף שני קודקודים מדרגה אי-זוגית והשאר בעלי דרגה זוגית. הוכחה דומה.
- גרף <u>מכוון</u> וקשיר מכיל מעגל אויילר $\Leftrightarrow$ $\forall v\in V: d_{in}(v)=d_{out}(v)$ (דרגות היציאה והכניסה של כל צומת שוות).
- גרף <u>מכוון</u> וקשיר מכיל מסלול אויילר $\Leftrightarrow$ קיימים שני צמתים u,v כך ש: $d_{in}(u)=d_{out}(u)+1, d_{in}(v)+1=d_{out}(v)$ ולכל צומת $w\neq u,v$ מתקיים: $d_{in}(w)=d_{out}(w)$ (ברור למה).

<u>*DFS: Depth First Search*: חיפוש לעומק</u>:

בשונה מה-*BFS*, בעת מציאת שכן של צומת, הא נמשיך לשאר שכניו אלא נמשיך לפתח את אותו שכן. רק כאשר יתקע יחזור אחורה וימשיך לשאר.

שתי פרוצדורות:

- *DFS(G)*: מנהלת את החיפוש ע"י מעבר על כל צמתי הגרף, ועבור כל צומת שלא נתקלנו בו כבר (לפי שדה צבע), מבצעת ממנו חיפוש חדש.
- *DFS-Visit(u)*: פרוצ' רקורסיבית המבצעת חיפוש בניסיון להגיע לכל מה שנגיש מצומת ההתחלה הנוכחי שעדיין לא ביקרנו בו.

<u>סיבוכיות</u>: $O(|V|+|E|)$.

<u>שדות</u>:

- **צבע**: לבן – טרם נתקלנו בצומת; אפור – נתקלנו אך טרם סיימנו (לא ביקרנו בכל צאצאיו); שחור – סיימנו את הביקור בצומת.
- $time$: מונה המתאר את סדרת האירועים שהאלג' נתקל בהם. בכל פעימה: נתקלים בצומת חדש / מסיימים טיפול בצומת.
- $\boldsymbol{d[u]}$: הזמן בו ניתקלנו ב-u לראשונה (*discovery*).
- $\boldsymbol{f[u]}$: הזמן בו סיימנו את הטיפול ב-u (*finish*). תמיד יתקיים: $f[u]>d[u]$.
- $\boldsymbol{\pi[u]}$: הצומת הקודם שגילה את u.

האלג' משתמש ברשימת צמתים ללא הנחות על סדר כלשהו, לכן שינוי סדר הצמתים ברשימה יביא לפלט שונה.

מצביעי π יוצרים אוסף של עצים מכוונים לכיוון השורש, המכונה יער ה-*DFS*. מצביעי π ההפוכים הם כיווני הקשתות המקוריים בגרף.

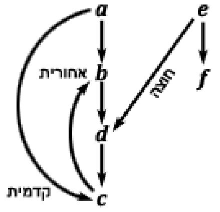
<u>תכונות</u>:

(1) עקרון הקינון:

לכל צומת u מתאים אינטרוול זמן בו הוא פעיל: $\left[d[u],f[u]\right]$. לכל שני צמתים u,v מתקיים: או שהאינטרוולים שלהם זרים, או שאחד מוכל בשני (לא תתכן חפיפה חלקית). אם הם מקוננים, אזי הצומת v עם האינטרוול ה<u>מוכל</u> יהיה <u>צאצא של u</u>, לו האינטרוול המכיל, ביער ה-*DFS*. כיוון שהאלג' פועל ברקורסיה, ההוכחה ניתנת לפישוט ע"י תיאור החזקת הצמתים במחסנית.

(2) סווג קשתות הגרף ע"י *DFS*:

האלג' עובר על כל קשתות הגרף, ובחלקן משתמש לבניית עץ ה-*DFS* (הגדרת π). קשתות אלו יסומנו כ<u>קשתות עץ</u>. סיווג שאר הקשתות:



- <u>קשת קדמית</u>: מחברת אב קדמון לצאצא.
- <u>קשת אחורית</u>: מחברת צאצא לאב קדמון (או לאב עצמו, אך לא קשת עץ).
- <u>קשת חוצה</u>: מחברת שני צמתים שאינם צאצא/אב קדמון.

<u>סיווג הקשתות בגרף מכוון</u>:

מסתכלים על (u,v) ברגע בו u בודק אותה:

- אם v לבן, זו קשת עץ: כיוון שזהו רגע ביצוע הקריאה הרקורסיבית.
- אם v אפור, זו קשת אחורית: הצבעה לצומת במסלול הנוכחי הנבדק, כלומר לצומת שהוא אב קדמון של v.

- אם v שחור, זו קשת קדמית או חוצה: אנו נמצאים בזמן $d[u] < time < f[u]$ ונתון ש-$time < f[v]$. לכן המקרים הם:
  - $f[v] < d[u]$ (אינטרוולים זרים): u,v אינם צאצא/אב-קדמון, לכן זו קשת חוצה (מצומת "מאוחר" u לצומת "מוקדם" v).
  - $d[u] < f[v]$ (אינטרווים מקוננים): u אב קדמון של v, לכן זו קשת קדמית.

<u>סווג הקשתות בגרף לא מכוון</u>:

במקרה זה יש רק קשתות עץ או קשתות אחוריות, ביחס לזמן בו הקשת מתגלית לראשונה. נניח (u,v) מתגלית לראשונה ב-u:

- אם v לבן, זו קשת עץ.
- אם v אינו לבן, זו קשת אחורית: v התגלה כבר קודם, ו-$d[u] > d[v]$. v צריך להיות במחסנית לפני u, כלומר אפור, אחרת אם היה שחור אזי כבר היינו מגלים את כל שכניו, בין היתר את u. לפיכך v חייב להיות במחסנית ולכן v אב קדמון של u (לכן זו קשת אחורית).

(3) עקרון המסלול הלבן:

נניח שב-G (מכוון או לא) קיים מסלול $v_0 \to v_1 \to \cdots \to v_k$, ונניח שהקודקוד הראשון המתגלה במסלול ה-DFS הוא v_1, אזי כל שאר הצמתים יהיו צאצאים של v_0 בעץ ה-DFS. מניחים כי v_{k-1} צאצא של v_0 וההוכחה באינדוקציה:

v_{k-1} מתגלה אחרי v_0 ומסתיים לפניו. בין לבין, בודק את כל שכניו כולל את v_k. לכן ברגע זה או קודם v_k מתגלה. לפיכך מתקיים: $d[v_0] < d[v_k] < f[v_{k-1}] < f[v_0]$. מעקרון הקינון נובע ש-$v_k$ צאצא של v_0.

<u>**שימושים**</u>:

(1) מציאת רכיבי קשירות בגרף לא מכוון G:

מריצים DFS, וכל עץ ביער ה-DFS הוא רכיב קשירות. כל עץ הוא רכיב קשירות – ברור. כל רכיב קשירות C הוא עץ: אם v_0 הוא הצומת הראשון ב-C שה-DFS מגלה, אזי לפי עקרון המסלול הלבן כל צמתי C הם צאצאים של v_0 (קיים מסלול מקשר מ-v_0 לכל שאר צמתי C כי C הוא רכיב קשירות).

(2) בדיקת שייכות שני צמתים u,v לאותו רכיב קשירות:

אחרי הרצת DFS, נגיע לשורש עץ ה-DFS של כל אחד מהצמתים ע"י מצביעי π ונשווה שורשים.

(3) האם G (מכוון או לא) מכיל מעגל:

קיים מעגל ב-G $\Leftrightarrow$ קיימת קשת אחורית.

(4) מיון טופולוגי של גרף <u>מכוון</u> אציקלי (DAG):

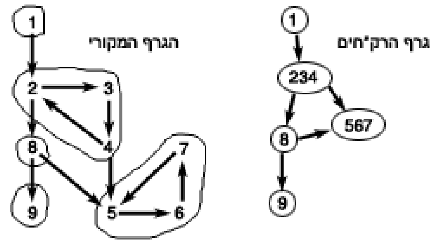
סידור צמתי G בסדר מלא $v_1, \ldots, v_n$ ($|V| = n$) כך שלכל קשת $(v_i, v_j) \in E$ מתקיים $i < j$. מיון טופולוגי אינו יחיד, התנאי היחיד הוא שהקשתות יהיו מכוונות קדימה. מציאת מיון טופולוגי: הרצת DFS, ולקיחת ערכי f בסדר <u>הפוך</u> (צומת בעל הערך הגבוה ביותר יהיה ראשון במיון).

בגרף G מכוון אין מעגל $\Leftrightarrow$ קיים לו מיון טופולוגי (הוכחה לפי סוג הקשת: מראים לכל סוג קשת (u,v) ש-$f(u) > f(v)$. כיוון שאין מעגל אז אין קשתות אחוריות).

(5) מציאת גשרים, קודקודים מנותקים ורכיבי דו-קשירות:

- **גשר**: קשת שאינה נמצאת על מעגל, ומחיקתה תנתק את הגרף (תהפוך אותו ללא קשיר).
- **קודקוד מנתק**: צומת שניתוקו יחד עם ניתוק כל הקשתות היוצאות ממנו תנתק את הגרף.
- **גרף דו-קשיר**: גרף קשיר שניתוק קשת כלשהי לא תהפוך אותו ללא קשיר (למשל מעגל). או: גרף קשיר ללא קודקודים מנתקים.

(6) מציאת רכיבי קשירות חזקה בגרף מכוון:



גרף מכוון הוא קשיר חזק אם מכל צומת ניתן להגיע לכל צומת אחר במסלול מכוון. אם לא, ניתן לפרק את V לרכיבים זרים א דמקסימלים, שהגרף המושרה מכל רכיב הוא קשיר חזק. רכיבים אלו הם <u>רק"חים</u>. לכל גרף מכוון ניתן להסתכל על <u>גרף הרקח"ים</u> – אותו גרף רק שכל צומת הוא רק"ח בגרף המקורי. גרף הרק"חים יהיה DAG: אם היה בו מעגל, אזי אותם רקח"ים המשתתפים במעגל היו באותו רק"ח שהיה מתגלה קודם.

<u>דוגמא</u>: נתון G מכוון, רוצים לבדוק האם הוא קשיר חזקה.

פתרון: נקבע שרירותית צומת s כלשהו ונבדוק:

- האם ניתן להגיע מ-s לכל צומת אחר: הרצת DFS מ-s ובדיקה שקיבלנו עץ אחד ביער ה-DFS. אם לא קיבלנו עץ אחד בלבד, נשתמש בעקרון המסלול הלבן: אם u אינו בעץ (כלומר עדיין לבן) אך קיים מסלול מ-s אליו, זהו מסלול לבן ולכן u צאצא של s.
- האם ניתן להגיע מכל צומת אחר ל-s: נבנה את הגרף ההפוך $G^t = (V, \{(v,u) | (u,v) \in E\})$, נריץ עליו DFS ונבדוק שמתקבל עץ אחד.

אם הראנו את שני הנ"ל, אזי יש מסלול מכל $u \in V$ לכל $v \in V$ - פשוט עוברים דרך s.

<u>חישוב רק"חים בגרף מכוון G</u>:

- מריצים DFS על G ומסדרים את הצמתים לפי סדר f יורד, נסמן רשימה זו L.
- בונים את הגרף ההפוך G^t.
- מריצים DFS על G^t כאשר הפרוצדורה הראשית משתמשת ב-L בתור רשימת הצמתים.
- כל עץ ב-DFS השני הוא רק"ח (אם נהפוך את כיוון הקשתות שוב לכיוון המקורי שב-G).

<u>טענות ותרגילים (מהתרגול, קשור לנקודה (5) לעיל)</u>:

- קשת (בגרף לא מכוון) אינה גשר ⇔ היא נמצאת על מעגל פשוט בגרף.

אם (u,v) אינה גשר, אזי לאחר הסרתה נשאר מסלול בין u ל-v, לכן היא נמצאת על מעגל פשוט (מסלול פשוט בין u ל-v בנוסף לקשת (u,v)).

- גרף לא מכוון וקשיר ניתן לכיוון לגרף קשיר בחוזקה ⇔ אין בו גשרים.

⇐: נסתכל על קשת (u,v) לאחר שכיוונו אותה. כיוון שהגרף המכוון החדש קשיר בחוזקה, יש מסלול גם מ-v אל u, ולכן (u,v) נמצאת על מעגל בגרף המכוון. הטענה הקודמת נובע שאינה גשר, וזה תקף לכל קשת בגרף. לכן בגרף אין גשרים.

⇒: מריצים DFS על הגרף, וכיוון שאינו מכוון מקבלים קשתות עץ וקשתות אחוריות בלבד. נכוון קשתות עץ מהורה לבן וקשתות אחוריות מצאצא לאב קדמון. נסמן את שורש העץ ב-s, ונראה שמ-s ניתן להגיע לכל צומת, ומכל צומת ל-s (וכך: מכל צומת ניתן להגיע לכל צומת). מסלול מ-s לצומת: על קשתות העץ. מסלול מצומת ל-s: מניחים באינדוקציה על צומת u ומוכיחים על צומת v הנמצאת עמוק יותר בעץ. לפי הנחה, אין גשרים ולכן כל קשת יושבת על מעגל. לפיכך קיימת קשת אחורית מ-v או צאצא שלו אל u או אב קדמון שלו – ולכן מ-v ניתן להגיע לצומת שלפי הנחת האינדוקציה מתחבר ל-s.

- נתונה $S\subseteq V$. רוצים לבדוק האם קיים מסלול (לא בהכרח פשוט) העובר בכל הצמתים ב-S.

מוצאים את גרף הרק"חים של G ומיון טופולוגי לגרף זה. בתוך כל רק"ח קיים מסלול בין כל הצמתים מ-S הנמצאים בו. לפי סדר המיון הטופולוגי של הרק"חים, מוצאים ע"י DFS / BFS מסלול בין כל C_i ל-C_{i+1} (רכיבי הקשירות).

- נתון גרף מכוון G, רוצים למצוא קבוצה מינימלית S כך שלכל $v\in V$ קיים $s\in S$ כך שקיים מסלול מ-s אל v.

מוצאים גרף רק"חים ל-G. לכל C (רכיב קשירות ב-G וצומת בגרף הרק"חים) <u>שהוא מקור</u> (אין קשתות שנכנסות אליו), בוחרים $c\in C$ ומכניסים אותו ל-S. מבצעים מיון טופולוגי על הרק"חים ומוכיחים באינדוקציה על C_i.

- אלגוריתם למציאת קודקודים מנתקים:

אלגוריתם: נגדיר לכל צומת שדה $low[u]$ שהוא: כמה גבוה בעץ ה-DFS ניתן להגיע מצומת u או צאצא שלו ע"י קשת אחורית (המרחק המינימלי מהשורש אליו ניתן להגיע). נריץ DFS, ולכל קודקוד שסיימנו לחשב את ערך ה-f שלו (סיים טיפול), נגדיר: $low[u]=\min\{d[u],\min\{d[v]|(u,v)-\text{קשת אחורית}\},\min\{low[w]|\ w\text{ בן ישיר של }u\}\}$. מתקיים: u (לא שורש) מנתק ⇔ קיים w <u>בן</u> של u כך ש-$low[w]\geq d[u]$.

הוכחה: אם קיים w כזה, אזי לא ניתן להגיע ממנו להורה של u, לכן u מנתק. אם u מנתק, אז קיים w כנ"ל, אחרת מכל בן (או צאצא) של u ניתן להגיע לאב כלשהו של u, כלומר הסרת u לא ניתקה את הגרף, בסתירה.

- טענה: u מנתק ⇔ יש לו לפחות שני בנים.
- טענה: לכל 2 רכיבי דו-קשירות יש לכל היותר קודקוד אחד משותף.

הוכחה: נניח בשלילה שלשני רכיבי דו-קשירות C_1,C_2 יש שני קודקודים משותפים u,v. נסתכל על $C_1\cup C_2$. אף צומת שנמצא רק באחד מהם אינו קודקוד מנתק כיוון ש-C_1,C_2 רכיבי דו-קשירות (מהגדרה). אם כן, אם ננתק את u, C_1,C_2 לא יתנתקו – מחוברים ע"י v. באופן דומה בניתוק v. לפיכך $C_1\cup C_2$ היה צריך להיות רכיב דו-קשירות אחד מלכתחילה כיוון שהוא תת גרף קשיר ללא קודקודים מנתקים.

- מסקנה: רכיבי דו קשירות מהווים חלוקה של קשתות הגרף.

הוכחה: כל קשת בפני עצמה היא תת-גרף דו-קשיר ולכן שייכת לרכיב דו-קשירות. מהטענה לעיל נובע שאין קשת משותפת (=2 צמתים משותפים) לשני רכיבי דו-קשירות שונים.

MST: Minimum Spanning Trees: עצים פורשים מינימליים (עפ"מ):

הקלט תמיד יהיה גרף לא מכוון וקשיר עם פונקציית משקלות על הקשתות $w: E \to \mathbb{R}$.

- **עץ פורש**: תת גרף של G שמכיל את כל צמתי הגרף והוא עץ (מורכב מ-$|V|-1$ קשתות מ-E). תמיד קיים כזה כי G קשיר.
- **משקל עץ פורש**: סכום משקלות קשתות העץ שהוא $w(T) = \sum_{e \in T} w(e)$.

נרצה למצוא עץ פורש מינימלי: עץ פורש בעל סכום משקלות מינימלי. עץ זה אינו יחיד בהכרח (למשל גרף בעל ערך w קבוע לכל קשת). אם יש משקלות שליליים, ניתן להוסיף להם $A > 0$ כלשהו וכך משקל כל עץ פורש יגדל בדיוק ב-$A \cdot (|V|-1)$, ונקבל שעפ"מ עכשיו $\Leftrightarrow$ עפ"מ קודם.

גישה חמדנית (*Greedy*):

נוסיף לעץ (מאותחל לריק) קשתות מ-G ונקפיד שלא יווצר מעגל. אם נצליח לבצע זאת $|V|-1$ פעמים נקבל עץ פורש. בדרך נבחר את הקשתות הקטנות ביותר שנוכל (בכל צעד תבחר הקשת המינימלית האפשרית).

גישה כללית:

- **קבוצה מבטיחה**: תת קבוצה $A \subseteq E$ כך שקיים עפ"מ המכיל את A. A לא חייבת להיות קשירה, אך ברור שאין בה מעגל. $\emptyset$ קב' מבטיחה.
- **קשת בטוחה**: אם A קב' מבטיחה ו-$e \notin A$, אזי e קשת מבטיחה אם $A \uplus \{e\}$ עדיין קב' מבטיחה.

האלגוריתם:

מאתחלים את A להיות הקבוצה הריקה. רצים בלולאה: כל עוד A אינו עץ פורש ($|A| \neq |V|-1$), הוסף קשת בטוחה e ל-A.

נכונות:

- קבוצה ריקה תמיד מבטיחה, לכן האתחול נכון.
- תמיד קיימת קשת בטוחה e ל-A קב' מבטיחה, כיוון ש-$A \subseteq T$, וכל $e \in T, e \notin A$ היא בטוחה מהגדרה.
- לבסוף: אם A עץ פורש מוכל בעפ"מ, אזי A הוא עפ"מ בעצמו.

כל רכיב קשירות בגרף (V, A) הוא עץ שיכול להיות גם ללא קשתות. באתחול גרף זה יהיה בעל $|V|$ רכיבי קשירות. אם e קשת בטוחה ל-A, אז הוספתה ל-A תחבר בין שני עצים (זרים בצמתים כי אין מעגלים) ותקטין את מספר רכיבי הקשירות באחד. לאחר $|V|-1$ צעדים נקבל רכיב קשירות אחד, שהוא העץ הפורש.

- **חתך**: פירוק של V לשתי קבוצות לא ריקות זרות כך ש-$V = V_1 \uplus V_2$.
- **קשת חוצה**: חתך היא קשת המחברת צומת ב-V_1 לצומת ב-V_2.
- **חתך מכבד**: את A (קב' מבטיחה) אם אין ב-A קשת חוצה (כל רכיב קשירות ב-A מוכל או ב-V_1 או ב-V_2).
- **קשת חוצה קלה**: בהינתן חתך, קשת תהיה חוצה קלה אם יש לה משקל מינימלי מבין כל הקשתות החוצות.

טענה:

תהא A קבוצה מבטיחה, יהא (V_1, V_2) חתך המכבד את A, ותהא $e = (u, v)$ קשת חוצה קלה של החתך. אזי e קשת בטוחה עבור A.

הוכחה: ידוע ש-$A \subseteq T$, T עפ"מ. אם $e \in T$, סיימנו. אם $e \notin T$: כיוון ש-T קשיר, קיימות קשתות ב-T החוצות את החתך (אחרת T לא קשיר). אם כן קיים מסלול π כלשהו בין u ל-v, ב-T, עם קשת חוצה e'. נחליף את e' ב-e ונסמן את העץ החדש T'. כיוון ש-e חוצה קלה: $w(e) \le w(e')$. T' עץ: יש בו $|V|-1$ קשתות; כמו כן מקיים: $w(T') = w(T) \underbrace{-w(e') + w(e)}_{\le 0} \le w(T)$. כיוון ש-$T$ עפ"מ אזי גם T' עפ"מ, ולכן $A \cup \{e\}$ בטוחה.

מסקנה: יהי C רכיב קשירות של A, אזי החתך $(C, V \backslash C)$ מכבד את A, וקשת חוצה קלה בחתך זה בטוחה ל-A.

אלגוריתם *Kruskal* למציאת *MST*:

אלגוריתם זה משתמש במבנה נתונים *Union-set*. באיתחול כל צומת הוא *set* בפני עצמו, ורצים על הקשתות לפי משקל לא יורד. בכל שלב: אם הקשת מחברת בין שני רכיבי קשירות (*sets*) שונים, נוסיפה ל-A ונאחד את שני ה-*sets* שחיברה ל-*set* אחד. אם לא מחברת בין שניים שונים, נדלג עליה (מדלגים על קשתות שסוגרות מעגל).

נכונות האלג':

- משקל מינימלי של e בחיבור שני רכיבי קשירות שונים: אם היתה e' קלה יותר, אז בדקנו אותה קודם (בודקים לפי סדר משקלות לא יורד), ולפיכך נפסלה ולכן סגרה מעגל. אם כן גם בתמונה הנוכחית תסגור מעגל, ולא ניתן לבחור אותה.
- e בטוחה ל-A: נניח שמחברת C_1 ל-C_2 (שני רכיבים קשירות שונים). נסתכל על החתך $(C_1, V \backslash C_1)$: e חוצה חתך זה, ולפי סדר הבדיקה היא הקלה מבין כל הקשתות החוצות חתך זה – מכאן שהיא חוצה קלה. לכן לפי הטענה לעיל היא בטוחה ל-A.

סיבוכיות: למימוש יעיל $O(|E| \cdot \log|V|)$.

אלגוריתם *Prim* למציאת *MST*:

מתחזקים רכיב קשירות אחד C (מאותחל ל-$\emptyset$). בכל שלב מוצאים את הקשת הקלה ביותר המחברת צומת ב-C לצומת במשלים של C – אותו צומת במשלים עובר לתוך C. האלגי לא פועל על קשתות אלא על צמתים: כל צומת במשלים $(V\backslash C)$ מחזיק *key* של משקל הקשת המינימלי בינו לבין מישהו ב-C. מחזיקים את כל הצמתים (המשלים של C) בתור עדיפויות. בכל שלב מוצאים את הצומת עם ה-*key* המינימלי (*extract-min*) ומוסיפים אותו ל-C, מעדכנים את ה-*keys* של שכניו (הנמצאים במשלים) ע"י *decrease-key*.

נכונות:

מיידית מהטענה: אם נסתכל על החתך $(C, V\backslash C)$ (כאשר $V\backslash C$ הוא תור העדיפויות), אזי כל צומת שמוציאים מ-$V\backslash C$ מחובר בקשת חוצה קלה ל-C, לכן זו קשת בטוחה.

סיבוכיות:

$O(|V|\cdot\log(|V|))$ להוצאות מהתור, $O(|E|\cdot\log(|V|))$ הפחתות מפתח בערימה בינארית או $O(|E|)$ בערימת פיבונצ'י. סה"כ:

$O(|E|\cdot\log(|V|))$ בערימה בינארית או $O(|V|\cdot\log(|V|)+|E|)$ בערימת פיבונצ'י – חיסכון לעומת *Kruskal* ל-E גדול.

תכונות עפ"מים (נובעים מאלג' *Kruskal*):

פלט האלג' של קרוסקל תלוי בסדר מיון הקשתות. אם לכל הקשתות משקלים שונים, האלג' מייצר עץ יחיד.

- טענה: האלג' של קרוסקל מסוגל לייצר כל עפ"מ אפשרי (מכאן: אם המשקלים שונים, יש עפ"מ יחיד)
- טענה: תהי $f:\mathbb{R}\to\mathbb{R}$ עולה ממש, ו-G עם פונ' משקלות w על הקשתות. נגדיר $w^*=f\circ w$ ומתקיים: T עפ"מ לפי $w \Leftrightarrow T$ עפ"מ לפי w^*.

הוכחה: עבור T עפ"מ מסוים, נסדר את הקשתות ב-G לפי סדר w עולה כך שקרוסקל יבנה את T. אותו סדר בדיוק יהיה עבור w^*.

- טענה: עפ"מים שונים נבדלים בקשתות שלהם אך לא במשקל קשתותיהם: יהיו T_1, T_2 שני עפ"מים שונים. אזי אם נסדר את משקל קשתותיהם לפי סדר עולה: $x_1,\dots,x_{|V|-1}$ ו-$y_1,\dots,y_{|V|-1}$ אזי: $\forall 1\le i\le |V|-1: x_i=y_i$.

הוכחה: נניח בשלילה שלא מתקיים, וקיים אינדקס j כך שבה"כ: $x_j<y_j$. נגדיר פונ' עולה ממש: $f(x)=\begin{cases}x, & x\le x_j\\ x+1, & x>x_j\end{cases}$. לפי המשקלות החדשים מתקיים: $x_1\le\cdots\le x_j\le x_{j+1}(+1)\le\cdots\le x_{|V|-1}(+1)$ - ה-"+1" יהיה אם המשקלים הללו שונים מ-x_j; ועבור y: $y_1\le\cdots\le y_j\le y_{j+1}+1\le\cdots\le y_{|V|-1}1$. כלומר: המשקל הכולל של העץ הראשון ישתנה לכל היותר ב-$|V|-1-j$, והשני ישתנה בדיוק ב-$|V|-j$, לכן בכל מקרה משקלם יהיה שונה בסתירה לשמירת עפ"מ תחת f עולה ממש.

טענות ותרגילים (מהתרגול):

- נתון גרף מכוון וקשיר עם פונ' משקלות על הקשתות. כל קשת צבועה באדום או כחול. למצוא עפ"מ עם מספר מקסימלי של קשתות אדומות.

פתרון: נשנה את אחד האלג' למציאת עפ"מ באופן הבא: השוואת קשתות ע"י w תוגדר: e_1 קלה מ-e_2 אם: או $w(e_1)<w(e_2)$ או $w(e_1)=w(e_2)$, e_1 אדומה ו-e_2 כחולה.

הוכחה: נגדיר פונ' משקלות חדשה w': אם e כחולה אז $w'(e)=w(e)$ ואם e אדומה אז $w'(e)=\varepsilon$ כך ש-ε קטן מספיק כדי שההשוואה בין הקשתות תהיה שקולה להגדרה בפתרון. כמו כן נניח ε קטן מספיק כך שלכל שני עצים פורשים: $w(T_1)<w(T_2)\Leftarrow w'(T_1)<w'(T_2)$. העץ T שהפתרון מחזיר מביא למינימום את $w'(T)$, בגלל שקילות הפתרון להגדרת w'. כעת נותר להראות:

א. T הוא עפ"מ לפי w: אם קיים T_1 כך ש-$w(T_1)<w(T)$ אזי $w'(T_1)<w'(T)$ בסתירה לכך ש-T עפ"מ לפי w'.

ב. לכל עפ"מ אחר T_1 לפי w יש מספר קטן או שווה של קשתות אדומות: $w(T)-\varepsilon=w'(T)\le w'(T_1)=w(T_1)-\varepsilon$

מסלולים קצרים (קלים) ביותר – מק"בים:

נתון גרף G מכוון או לא, ופונ' משקלות על הקשתות $w:E\to\mathbb{R}$ (יכול להיות שלילי). משקל מסלול = סכום המשקלות שעל קשתות המסלול. לזוג צמתים *u,v* נגדיר $\delta(u,v)$ = המשקל המינימלי של מסלול המחבר את u ל-v (אם קיים; ∞ אחרת).

בעיות *single source* או *shortest paths*: עבור צומת התחלה s נרצה לחשב את כל הגדלים $\delta(s,u)$ לכל $u\in V$ (להבדיל מבעיית *All pairs S.P.*).

הנחה: אין מעגלים שליליים (סכום משקל הקשתות במעגל שלילי) נגישים מ-s. אם היו, אזי כל הצמתים u שיש מסלול אליהם מ-s העובר במעגל היו מקיימים $\delta(s,u)=-\infty$. מעגלים במשקל 0 הם תקינים, פשוט ניתן להחליט לעבור בהם מספר כלשהו של פעמים.

אם אכן אין מעגלים שליליים אזי: תמיד קיים מק"ב <u>פשוט</u> (לא עובר באותו צומת פעמיים) מ-s לכל u הנגיש אליו.

<u>תכונות מק״בים</u>:

- תת-מסלול של מק״ב הוא מק״ב.
- אשמ״ש: אם $(u,v)\in E$ אזי $\delta(s,v)\le\delta(s,u)+w(u,v)$.
- אם u צומת אחד לפני v על מק״ב מ-s ל-v אז מתקיים השוויון: $\delta(s,v)=\delta(s,u)+w(u,v)$.
- אם $w\equiv 1$ אזי משקל המסלול שווה למספר הקשתות על המסלול, והבעיה נפתרת ע״י *BFS*.

באופן דומה ל-*BFS*, יוחזקו השדות הבאים לכל צומת:

- $d[u]$: משקל המק״ב מ-s ל-u שהתגלה עד כה. מאותחל ל-∞ ($d[s]$ מאותחל ל-0).
- $\pi[u]$: מצביע לצומת הקודם על המק״ב שהתגלה עד כה. מאותחל ל-*null*.

<u>אלגוריתם כללי</u>:

פרוצדורת העדכון *Relax*: הפעלת *relax* על (u,v) בודקת אם $d[v]>d[u]+w(u,v)$. אם כן, נעדכן את $d[v]$ ואת $\pi[v]\leftarrow u$.

לאחר איתחול עוברים על הקשתות בסדר כלשהו כל עוד יש *relax* משפר. כשאין *relax* משפר עוצרים. אלג׳ זה לא יעצור אם יש מעגל שלילי נגיש מ-s. אם אין מעגלים שליליים, האלג׳ תמיד יעצור ובסופו יתקיים לכל u: $d[u]=\delta(s,u)$.

לכל u: $d[u]$ יורד מונוטונית במהלך האלג׳; תמיד מתקיים $d[u]\ge\delta(s,u)$.

<u>עץ המק״בים</u>:

טענה: בכל רגע שהוא, מצביעי ה-π שאינם *null* (אך כולל s) יוצרים עץ מכוון כלפי השורש, והשורש הוא s.

הוכחה: הגרף המורכב ממצביעי π הוא קשיר וחסר מעגלים, ולכל u שאינו מצביע ל-*null* מסלול יחיד לכיוון s כיוון ששדה π יחיד.

- אם המסלול מסתיים ב-u_k כך ש-$\pi(u_k)=null$ אזי $u_k=s$. נניח שלא, אז נסתכל על ה-*relax* שנעשה על (u_{k-1},u_k) שהביא ל-$\pi(u_{k-1})\leftarrow u_k$: $d(u_k)+w(u_{k-1},u_k)<d(u_{k-1})$. מכאן ש-$d(u_k)<\infty$ (כי ה-*relax* נעשה). אם אכן $u_k\ne s$, אז הפעם הראשונה ש-$d[u_k]$ מקבל ערך סופי הוא ב-*relax*, ואז $\pi[u_k]$ מקבל ערך שאינו *null* ולא יקבל *null* לעולם, בסתירה לכך ש-$\pi[u_k]=null$. לכן $u_k=s$.
- אין מעגל: נניח שיש מעגל, באמצעות חיבור אי-שוויונים נקבל שהמעגל הוא שלילי.

<u>בעץ המק״בים מתקיים</u>:

- כל צומת בעץ הוא צומת שנגיש מ-s בגרף המקורי: נניח u נגיש מ-s, אזי $d[u]=\delta(s,u)<\infty$ - $\pi[u]$ קיבל ערך שונה מ-*null* ולכן u בעץ.
- המסלול (ההפוך) מכל צומת לשורש s הוא מק״ב בגרף המקורי: מניחים באינדוקציה על עומק השורש k. מתקיים:
  אורך מסלול מסוים מ-s ל-$u_k=\sum_{i=1}^{k}w(u_{i-1},u_i)\le d[u_k]=\delta(s,u_k)$, ולכן אותו מסלול מסוים הוא מק״ב.

<u>חישוב מק״בים ב-*DAG*</u>:

מבצעים מיון טופולוגי על G, וניתן להתעלם מכל הצמתים הקודמים ל-s במיון. נבצע איתחול, נעבור על כל צומת לפי סדר המיון הטופולוגי ונבצע *relax* על כל הקשתות היוצאות ממנו.

נכונות: כאשר הלולאה המרכזית מגיעה לצומת u, מתקיים $d[u]=\delta(s,u)$. אינדוקציה על סדר המיון הטופולוגי: נניח נכונות עד v כאשר $s\to\cdots\to v\to u$ הוא מק״ב, וכל הקודמים ל-u נמצאים לפניו במיון. לפי הנחה: $d[v]=\delta(s,v)$ כשמגיעים ל-v, ואז מבצעים *relax* על הקשתות כולל (v,u) ומתקיים לאחר מכן: $d[u]\le d[v]+w(v,u)\Leftrightarrow d[u]\le\delta(s,v)+w(v,u)$. אגף ימין הוא תת-מסלול של מק״ב + אורך הקשת (v,u), לכן סה״כ שווה ל-$\delta(s,u)$. כיוון שתמיד גם $d[u]\ge\delta(s,u)$ אזי: $d[u]=\delta(s,u)$.

<u>אלגוריתם של *Ford*</u>:

הקלט הוא גרף כללי, משקלות שליליים אפשריים (לא מעגלים שליליים). אתחול כרגיל. הלולאה: כל עוד קיימת קשת עם *relax* משפר, נבצע אותו. בסיום מתקיים $d[u]=\delta(s,u)$ לכל u. זמן הריצה יכול להיות נוראי: יתכן סדר *relax* שיגרום ל-2^n צעדים.

<u>אלגוריתם *Bellman-Ford*</u>:

אם אין מעגלים שליליים אזי תמיד קיים מק״ב פשוט, כלומר לכל היותר בעל $|V|-1$ קשתות. האלג׳ עושה $|V|-1$ איטרציות, כאשר בכל אחת הוא מבצע *relax* על כל הקשתות. סיבוכיות: $O(|V|\cdot|E|)$. הוכחה באינדוקציה על k מספר האיטרציה (של הלולאה הרצה עד $|V|-1$).

אם עדיין קיים *relax* משפר, החלק השני של האלגוריתם מגלה זאת ומחזיר *false* = קיימים מעגלים שליליים בגרף. הוכחה:

אם אין מעגל שלילי, ראינו שמתקיים בסוף לכל u: $d[u]=\delta(s,u)$. אם יש מעגל שלילי, נטען שיש *relax* משפר: אחרי $|V|-1$ האיטרציות כל ערכי ה-d קטנים מ-∞. אם נניח שאכן אין *relax* משפר נקבל אי-שוויונים מהצורה: $d[v_i]\le d[v_{i-1}]+w(v_{i-1},v_i)$ (v_i צמתים במעגל). אם נחבר את כולם יחד ונקבל: $0\le\sum_{i=1}^{k}w(v_i,v_{i+1})$ – מעגל אי-שלילי, בסתירה.

אלגוריתם של *Dijkstra*:

הנחה: משקלות אי-שליליים.

שימוש בתור עדיפויות Q המאותחל ל-V, וקבוצה S המאותחלת ל-$\emptyset$ שתחזיק את כל הצמתים שכבר הוצאנו מ-Q וחישבנו את ערך ה-δ שלהם. התור מנוהל לפי ערכי ה-d. לכל *extract-min* שמבוצע על Q, מבצעים *relax* על כל שכניו (הנמצאים ב-Q עדיין. *relax* על אלו שלא ב-Q אינו רלוונטי). אם ה-*relax* משפר, זה גורר *decrease-key* על אותו שכן הנמצא ב-Q.

ההבדל מ-*Bellman-Ford*: כאן כאשר צומת כלשהו מוצא מהתור, בטוח לא ישופר עוד ערך ה-d שלו. *B-F* מחכה כל הזמן לשיפורים נוספים.

סיבוכיות:

על כל צומת וכל קשת עוברים פעם אחת: $O(|E|+|V|)$ + פעולות על Q: $|V|$ פעולות *extract-min* + $\geq |E|$ פעולות *decrease-key*. סה"כ:

ערימה בינומית: $O\big((|E|+|V|)\cdot\log(|V|)\big)$, ערימת פיבונצ'י: $O(|V|\cdot\log(|V|)+|E|)$.

נכונות:

כאשר מוציאים v מהתור מתקיים $d[v]=\delta(s,v)$, הוכחה באינדוקציה על סדר ההוצאה מהתור: s – ברור. נניח הבא לצאת הוא v. נסתכל על המק"ב מ-s ל-v, שחלקו צמתים ב-S וחלקו ב-Q. נסמן את האחרון שב-S (כשיוצאים מ-s) ב-x, והראשון שלא ב-S ב-u. כאשר x יצא מ-Q מתבצע *relax* על (x,u) ומתקיים אי השוויון הרגיל: $d[u]\leq d[x]++w(x,u)=\delta(s,x)+w(x,u)=\delta(s,u)$ ולכן $d[u]=\delta(s,u)$. כיוון ש-v יוצא כעת מהתור, ו-u טרם יצא, יתקיים: $d[v]\leq d[u]$. כעת מתקיים:

$\delta(s,v)\leq d[v]\leq d[u]=\delta(s,u)$ ולכן $\delta(s,v)\leq\delta(s,u)$. כמו כן מתקיים: $\delta(s,v)=\delta(s,u)+w(u,v)$ ולכן $\delta(s,v)\geq\delta(s,u)$. מכאן שאי השוויון הראשון הוא בעצם שוויון, ובפרט: $d[v]=\delta(s,v)$.

תרגילים:

- G גרף מכוון עם $w: E\rightarrow\{0,1,2\}$. למצוא מקי"בים מ-$s\in V$.

פתרון: משתמשים בעובדה שסיבוכיות זמן ריצה של *Dijkstra* היא $O(|V|+|V|\cdot extract\_min+|E|\cdot decrease\_key)$. נממש תור עדיפויות בצורה יעילה יותר באופן הבא: כל המק"בים פשוטים (אין משקלות שליליים) ולכן משקל מסלול הוא לכל היותר $2\cdot(|V|-1)$. נחליף את ∞ ב-$2(|V|-1)+1$ ונחזיק מערך באורך זה, שיחזיק רשימות מקושרות דו כיווניות: בכל אינדקס i במערך יהיו כל הצמתים שערך ה-d שלהם הוא i. לצמתים גם יהיה מצביע למקום ברשימה המצביע אליהם חזרה. ואז מתקיים:

- *decrease key*: מחיקת מצביע מהרשימה בה נמצא במערך והשמת מצביע לצומת בראש הרשימה באינדקס החדש – $O(1)$.
- *extract min*: מציאת האיבר הראשון במערך שלא מכיל רשימה ריקה. נחזיק משתנה עזר להיכן הגענו, כי כל קריאה באה ל-*extract min* תחזיר ערך גדול או שווה לקודמו. סה"כ נעבור על כל הצמתים פעם אחת – $O(|V|)$ לכל הקריאות = $O(1)$ לקריאה.

סה"כ: $O(|V|+|E|)$.

- נתון G מכוון, $w: E\rightarrow R$, $s,t\in V$, כל קשת צבועה אדום / כחול. למצוא מק"ב מ-s ל-t בעל מספר זוגי של קשתות אדומות.

פתרון: ניצור גרף חדש G' באופן הבא: לכל צומת v ב-V יהיו שני צמתים v_0,v_1 ב-V'; לכל קשת $(u,v)\in E$ יהיו הקשתות:

- $(u_0,v_0),(u_1,v_1)$ אם (u,v) כחולה.
- $(u_0,v_1),(u_1,v_0)$ אם (u,v) אדומה.

נריץ *Dijkstra* מ-s_0 ונחזיר את המק"ב מ-s_0 ל-t_0.

נכונות: מההגדרות נובע שקיים מסלול מ-s_0 ל-t_0 אמ"מ קיים מסלול מ-s ל-t באותו אורך ובעל מספר זוגי של קשתות אדומות, בפרט מק"ב.

מק"בים בין כל זוגות הצמתים – *All Pairs Shortest Paths*:

פתרונות נאיביים:

- אם יש משקלות שליליים (לא מעגלים שליליים), נריץ *BF* $|V|$ פעמים, כל פעם עם צומת אחר כצומת התחלה. זמן ריצה: $O(|V|^2|E|)$ ויכול להגיע עד $O(|V|^4)$.
- אם אין משקלות שליליים נבצע *Dijkstra* $|V|$ פעמים וזמן הריצה יהיה $O(|V|^2\cdot\log(|V|)+|V|\cdot|E|)$ שזה לכל היותר $O(|V|^3)$.

נשתמש ביצוג מטריציאלי של הגרף, כאשר לצמתים מספרים סידוריים 1 עד n.

- **מטריצה** W: $W_{i,j}=\begin{cases}0, & i=j\\ \infty, & (i,j)\notin E\\ w(i,j), & (i,j)\in E\end{cases}$ (מטריצת משקלות).

- **מטריצה** D: מטריצה שתחזיק את אורכי המסלולים הקצרים ביותר שהתגלו עד כה בין כל i לכל j. אתחול אפשרי: לקחת את W.
- **מטריצה** π: מטריצת המצביעים במסלול הטוב ביותר שהתגלה עד כה. איתחול: $\pi_{i,j}=\begin{cases} null, & i-j \\ null, & w_{i,j}=\infty \\ i, & w_{i,j}<\infty \end{cases}$.

אלגוריתם ראשון (כמו B-F):

נסמן את מטריצת התשובה (מטריצת המק״בים) ב-D', ונגדיר: $\forall i,j: D'_{i,j}=\min\{D'_{i,j},\min_{1\le k\le n}\{D_{i,k}+W_{k,j}\}\}$. נגדיר פעולת ״כפל״ מטריצות: $D'=D\circ W$, כאשר במקום ״+״ נשתמש ב-״min״, ובמקום ״·״ נשתמש ב-״+״ (למשל: $r_1c_1+\dots+r_nc_n \to \min\{r_1+c_1,\dots,r_n+c_n\}$).
הוכחנו שמספיק לבצע $|V|-1$ איטרציות ב-B-F, ולכן מטריצת המק״בים תהיה $D^{|V|-1}=\underbrace{W\circ W\circ\ldots\circ W}_{|V|-1\ times}$.

סיבוכיות:
קל להוכיח ש״כפל״ המטריצות שהוגדר לעיל אסוציטיבי ולכן ניתן להשתמש ב-$\log n$ הכפלות במקום ב-n הכפלות. נקבל: $O(|V|^3\cdot\log(|V|))$.
הערה: האתחול נחשב לאיטרציה ראשונה, לכן נבצע ״כפל״ רק $|V|-2$ פעמים בפועל.

הערות:
1. **מציאת מעגלים שליליים**: כמו B-F, לאחר סיום כל ההכפלות, נבצע הכפלה נוספת ונבדוק האם היא משפרת $D_{i,j}$ כלשהו. אם כן אז יש מעגל שלילי. אופציה נוספת: אחרי סיום (כל ה-$|V|-1$ איטרציות) נבדוק אם יש ערך שלילי על האלכסון.
2. **עדכון מטריצת** π: אם איזשהו $relax$ שיפר, ניקח את אותו ערך k אליו שיפרנו ונבצע $\pi_{i,j}\leftarrow k$.
3. **עדכון** π **בגרסה המהירה**: עבור ה-k מה-$relax$ המשפר נבצע את ההשמה: $\pi_{i,j}\leftarrow\pi_{k,j}$.
4. **מקום**: אין צורך בעותקים של D. באופן נאיבי מספיק להחזיק שניים: כל תוצאה חדשה נשים בעותק האחר. ניתן גם עם עותק אחד.

תרגילים:

- נתונים n סוגי מטבעות $c_1,\dots,c_n$ ומטריצת המרה בה: כמה מטבעות מסוג j נקבל עבור מטבעות מסוג $i=(a_{ij})$. רוצים לבדוק האם קיים מסלול המרה שיביא לרווח, כלומר אינדקסים i_j כך ש: $a_{i_1,i_2}\cdot a_{i_2,i_3}\cdot\ldots\cdot a_{i_k,i_1}>1$.

פתרון: מוציאים log משני האגפים ומקבלים $-\log(a_{i_1,i_2})-\dots-\log(a_{i_k,i_1})<0$. נבנה גרף מכוון שלם שקודקודיו הם כל c_i ולכל קשת (c_i,c_j) ניתן את המשקל $-\log(a_{ij})$. כעת הבעיה שקולה למציאת מעגל שלילי בגרף, וזאת ניתן ע״י B-F. כיוון שהמעגל שלם, אז כל צומת נגיש מכל צומת ולכן ניתן לבצע B-F מכל צומת שהוא. זמן ריצה: $O(|V|\cdot|E|)=O(|V|^3)$.

- נתונה מערכת אי-שוויונים על המשתנים $x_1,\dots,x_n$ מהצורה: $x_i-x_j\le c_{ij}$. רוצים למצוא פתרון למערכת או להוכיח שלא קיים כזה.

פתרון: נבנה גרף שלם לא מכוון שצמתיו הם כל x_i ומשקל כל קשת (x_i,x_j) הוא c_{ij}. נוסיף לגרף צומת חדש s עם קשתות במשקל 0 לכל צמתי הגרף. נטען כי קיים פתרון למערכת אמ״מ אין מעגל שלילי בגרף שיצרנו:
אם נניח שיש מעגל שלילי, אזי סיכום אי השוויונות המתאימים יביאו לסתירה. אם נניח שאין מעגל שלילי, אז B-F מחזיר לכל x_i את $\delta(s,x_i)$.
נראה שהצבת ערכים אלו הם פתרון למערכת: לכל i,j צריך להתקיים $x_i-x_j\le c_{ij}$ וזה אמ״מ $\delta(s,x_i)-\delta(s,x_j)\le c_{ij}=w(x_j,x_i)$ ומכאן מתקבל אי שוויון בסיסי לגבי מק״בים.

- נתון גרף מכוון ללא מעגלים שליליים ופונ׳ משקלות. לכל v נגדיר: $\delta^*(v)=\min_{u\in V}\{\delta(u,v)\}$. למצוא את δ^*.

פתרון: נוסיף צומת חדש s עם קשתות במשקל 0 לכל צמתי הגרף. נריץ B-F ונחזיר לכל $v\in V$ את $\delta(s,v)$.
נכונות: מתקיים ש-$\delta(s,v)=\min\{w(s,u)+\delta(u,v)\}=\min\{\delta(u,v)\}=\delta^*(v)$.

- נתון גרף מכוון עם פונ׳ משקלות וללא מעגלים שליליים. רוצים להחזיר $True$ אמ״מ קיים מעגל במשקל 0 בגרף.

פתרון: נוסיף לגרף צומת חדש s עם קשתות במשקל 0 לכל צמתי הגרף. נריץ B-F מ-s ונבנה מתוצאותיו את גרף המק״בים G' (בו קשת מופיעה רק אם היא חלק ממק״ב ב-G). בעזרת DFS נבדוק אם יש קשת אחורית = יש מעגל מכוון. אם כן, נחזיר T אחרת F.
נכונות: אם יש מעגל באורך k ב-G' אז סיכום השוויונות $\delta(s,v_i)=\delta(s,v_{i-1})+w(v_{i-1},v_i)$ (כאשר $v_0=v_k$) יביא ל: $\sum_{i=1}^{k}w(v_{i-1},v_i)=0=w(\text{G המעגל ב})$, כלומר קיים ב-$G$ מעגל במשקל 0. אם קיים ב-G מעגל במשקל 0 אז אם נניח בשלילה שאינו ב-G' אז קיימת קשת מהמעגל שאינה ב-G' ונקבל אי שוויונות מהצורה $\delta(s,v_i)\le\delta(s,v_i)+w(v_i,v_{i+1})$ פרט לקשת אחת (בה״כ (v_k,v_1)): $\delta(s,v_1)\underset{\text{קטן ממש}}{<}\delta(s,v_1)+w(v_1,v_2)$. מסיכום אי השוויונות נקבל כי $0<w(\text{G המעגל ב})$, בסתירה.

אלגוריתם של *Floyd-Warshall*:

ממספרים באופן שרירותי את הצמתים מ-1 עד n. עבור מק"ב מצומת i לצומת j, נסתכל על כל הצמתים הפנימיים, ונגדיר את גובה המק"ב k כאינדקס המקסימלי מבין הצמתים הפנימיים באותו מק"ב. בונים מק"בים לפי סדר גובה עולה: לכל i,j,k נחשב את אורך המק"ב מ-i ל-j בגובה $\geq k$ החל מ-$k=0$. בסיום נקבל את כל המק"בים.

- $D_{i,j}^{(k)}$: אורך המסלול הקצר ביותר מ-i ל-j בגובה $\geq k$.
- $D^{(0)} = W$: מסלולים ללא צמתים פנימיים.
- בניית $D^{(k)}$ מתוך $D^{(k-1)}$:

$D_{i,j}^{(k)} = \min\left\{D_{i,j}^{(k-1)}, D_{i,k}^{(k-1)} + D_{k,j}^{(k-1)}\right\}$: או שהם שווים או האורך המינימלי של מסלול מ-i ל-j העובר דרך k ושאר הצמתים $\geq k$.

סיבוכיות: $O(|V|^3)$.

הערות:

1. **מקום**: מספיק מקום למט' אחת ועדכון כל ערך חדש נשים במקום הערך הישן.
2. **עדכון מטריצת π**: אם הערך שונה, כלומר קיים מסלול טוב יותר העובר דרך k (הגורם הימני): $\pi_{i,j}^{(k)} \leftarrow \pi_{k,j}^{(k-1)}$. אם לא (הגורם השמאלי) אז: $\pi_{i,j}^{(k)} \leftarrow \pi_{i,j}^{(k-1)}$.

אלגוריתם של *Johnson*:

רעיון האלגוריתם: הופך את כל המשקלות לאי-שליליים ומריץ את *Dijkstra* $|V|$ פעמים. הטרנספורמציה הנדרשת:

$h: V \to \mathbb{R}$, $w^*(u,v) = w(u,v) + h(u) - h(v)$. טענה: מק"ב תחת w $\Leftrightarrow$ מק"ב תחת w^*.

הוכחה: עבור מסלול כלשהו $w(\pi) = w(v_1,v_2) + \dots + w(v_{k-1},v_k)$ מתקיים: $w^*(\pi) = \dots = w(\pi) + h(v_1) - h(v_k)$ - כלומר ההבדל תלוי רק בצומת ההתחלה וצומת הסיום, ולא במסלול. $h(v_1) - h(v_k)$ תהיה תוספת קבועה עבור כל מסלול מ-v_1 ל-v_k ולכן מק"ב נשמר.

כעת נבנה h מתאימה:

נעתיק את הגרף לגרף חדש G', ונוסיף לו צומת s שיחובר לכל צמתי G' עם משקל *0* לכל (s,u). נבצע *B-F* אחד מ-s, ונגדיר: $h(u) = \delta_{G'}(s,u)$.

אם כל המשקלות הישנים גדולים או שווים ל-0, אז $h \equiv 0$ (כי ברור שהמק"ב מ-s לכל הצמתים יהיה הקשת הישירה שמשקלה 0, ואז $w = w^*$).

ההוכחה פשוטה: $w^*(u,v) = w(u,v) + h(u) - h(v) \geq 0$ אמ"מ: $w(u,v) + \delta(s,u) \geq \delta(s,v)$ - תכונה בסיסית של מק"ב.

זרימה ברשתות: *Network Flow*:

רשת זרימה:

גרף מכוון G עם צומת מקור $s \in V$ וצומת מטרה $t \in V$ ופונקצית קיבול על הקשתות $c: E \to \mathbb{R}^+$ ($c(e) > 0\ \forall e \in E$; לכל $e \notin E$: $c(e) = 0$).

מניחים כי כל צומת נגיש מ-s ו-t נגיש מכל צומת. בפרט, G קשיר ו-$|E| \geq |V| + 1$.

זרימה - *flow*:

פונקציה המוגדרת לכל זוג צמתים: $f: V \times V \to \mathbb{R}$. בפועל f מקבלת ערכים שאינם 0 רק על קשתות ואנטי קשתות. זרימה חוקית מקיימת:

- אילוצי קיבול: $\forall(u,v) \in V \times V: f(u,v) \leq c(u,v)$. אם $(u,v) \notin E$ אז $f(u,v) = f(v,u) = 0$.
- אנטי סימטריה: $\forall(u,v) \in V \times V: f(u,v) = -f(v,u)$ (מכאן: $f(u,u) = 0$).
- שימור הזרימה: לכל $u \neq s,t$ מתקיים $\sum_{v \in Adj(u)} f(u,v) = 0$ (ניתן להגיד לכל $v \in V$ כי זרימה של זוג שאינו קשת היא 0).

ערך:

ערך זרימה f המסומן $|f|$ הינו סה"כ הזרימה הנכנסת ל-t. **המטרה**: חישוב זרימה חוקית עם ערך מקסימלי $|f| = \sum_{u \in V} f(u,t)$.

תכונות וסימונים:

יהיו $A, B \subseteq V$ אזי $f(A,B) = \sum_{u \in A, v \in B} f(u,v)$. חוק שימור הזרימה: $f(\{u\},V) = f(u,V) = \sum_{v \in V} f(u,v) = 0$.

- לכל $X, Y \subseteq V$, f זרימה חוקית, מתקיים: $f(X,Y) = -f(Y,X)$.
- אם $X \cap Y = \emptyset$ אז: $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$ (ניתן גם להפוך את סדר הגורמים: $f(Z, X \cup Y) = \dots$).

אלגוריתם **Ford-Fulkerson**:

אתחול f ל-0 לכל (u,v). כל עוד קיים מסלול משפר, נגדיל את הזרימה f ע״י הזרמה נוספת לאורך המסלול.

<u>הרשת השיורית - Residual Network</u>:

נתונה רשת זרימה G עם פונ' קיבול c וצמתים s,t, וכן זרימה חוקית f. נסמן את הרשת השיורית G_f: הצמתים הם V והקשתות הן: $(u,v)\in E$ כך ש-$f(u,v)<c(u,v)$ (נותר פוטנציאל להזרים בהן) או אנטי קשתות (v,u) עבורן $f(u,v)>0$. קיבול שיורי: $c_f(u,v)=c(u,v)-f(u,v)$ - כמה עוד אפשר להזרים דרך אותה קשת. כאשר $f\equiv 0$ (באתחול) אז $G=G_f$.

$\Leftarrow$ מסלול משפר הוא מסלול מ-s ל-t ברשת השיורית G_f (קשתות עם קיבול שיורי 0 <u>לא</u> יופיעו ב-G_f).

אם p מסלול משפר, נסמן ב- $c_f(p)$ את הקיבול המינימלי של קשת של p = הגבול השיורי של p. נזרים ב-p זרימה נוספת בכמות $c_f(p)$ ונגדיר זרימה חדשה: $f^*(u,v)=\begin{cases}f(u,v), & (u,v)\notin p\\ f(u,v)+c_f(p), & (u,v)\in p\end{cases}$

<u>טענה</u>: f^* זרימה חוקית ו-$|f^*|=|f|+c_p>|f|$ (c_p סימון מקוצר ל-$c_f(p)$):

- מקיימת את אילוץ הקיבול.:
- מקיימת את שימור הזרימה.

כיוון שהאלג' לא מציין כיצד ניתן למצוא את המסלול מ-s ל-t, ניתן למצוא אותו ע״י *BFS*. מספר הצעדים (עבור קיבולים שלמים / רציונאלים שהפכנו לשלמים ע״י מכנה משותף) חסום מלעיל ע״י ערך הזרימה המקסימלית $|f^*|$. זאת כיוון שבכל צעד $c_p\geq 1$ ולכן ערך הזרימה גדל בערך ≥ 1 בכל צעד.

<u>משפט Min Cut Max Flow</u>:

נגדיר חתך כחלוקה של V לשתי קבוצות זרות (S,T) כך ש-$s\in S, t\in T$. הקיבול של החתך הוא $c(S,T)$ והזרימה דרך החתך היא $f(S,T)$.

<u>טענה</u>: לכל זרימה f ולכל חתך (S,T) מתקיים: $|f|=f(S,T)$.

<u>טענה</u>: לכל זרימה f ולכל חתך (S,T) מתקיים: $f(S,T)\leq c(S,T)$ (אם $(u,v)\in E$ כך ש-$u\in S, v\in T$ אז $f(u,v)\leq c(u,v)$, ואם היא אנטי קשת, כלומר $(v,u)\in E$ אזי $f(u,v)=-f(v,u)\leq 0=c(u,v)$).

<u>המשפט</u>:

התנאים הבאים שקולים:

1. f זרימה מקסימלית.
2. הרשת השיורית G_f לא מכילה מסלול מ-s ל-t.
3. קיים חתך (S,T) כך ש-$|f|=c(S,T)$.

הוכחה:

$1\Rightarrow 2$: מיידי, כיוון שאם היה מסלול משפר ב-G_f אז היינו יכולים להגדיל את ערך הזרימה.

$2\Rightarrow 3$: נניח שאין מסלול משפר ב-G_f, נבנה חתך (S,T) כך ש-S מכיל את כל הצמתים הנגישים מ-s ב-G_f ו-T את כל השאר ($t\in T$ כי לפי הנחה לא נגיש מ-s). כדי להראות את (3) נראה שלכל (u,v) בחתך מתקיים $f(u,v)=c(u,v)$ ($u\in S, v\in T$). נחלק לשני מקרים:

- (u,v) קשת מקורית בגרף: אם $f(u,v)<c(u,v)$ אזי $c_f(u,v)>0$ ולכן $(u,v)\in E_f$: כיוון ש-u נגיש מ-s, גם v נגיש, בסתירה.
- (u,v) אנטי קשת, כלומר (v,u) קשת. מתקיים $f(u,v)=f(v,u)=0$. אם לא יתקיים, אז $f(v,u)>0$ ו-$f(u,v)<0$ ומכאן $(u,v)\in E_f$ ו-v נגיש מ-s כמו קודם, בסתירה.

$3\Rightarrow 1$: מיידי, כי לכל זרימה מתקיים $|f|\leq c(S,T)$, לכן $|f|$ מקסימלית.

<u>יעילות (Ford-Fulkerson, גרסת Edmonds-Korp)</u>:

נרצה לחפש מסלולים משפרים <u>קצרים ביותר</u> מבחינת מס' הקשתות, ע״י *BFS* על הרשת השיורית. מס' האיטרציות של האלג' הוא מס' המסלולים המשפרים. מספר האיטרציות הוא $O(|E|\cdot|V|)$ ולכן סה״כ זמן הריצה (*BFS* בכל איטרציה) הוא $O(|E|^2\cdot|V|)$.

מספר האיטרציות חסום כך כיוון שבכל שיפור, קשת אחת לפחות על המסלול המשפר נהיית רוויה (אותה קשת שקיבולה השיורי הוא c_p). אותה קשת לא תהיה ברשת השיורית הבאה, אך יכולה לחזור לרשת השיורית שאחריה. סה״כ יכולה לחזור $\frac{|V|}{2}$ פעמים בערך. לכן סה״כ: $O(|E|\cdot|V|)$.

רשת שכבתית: *BFS* על G_f מ-s בונה שכבות, כאשר s נמצא לבדו בשכבה ה-0, t נמצא לבדו בשכבה האחרונה. שאר הצמתים נמצאים בין לבין לפי מרחקם מ-s. נחזיק רק קשתות מעבר משכבה לשכבה, וזו תהיה הרשת השכבתית.

אלגוריתם של דיניץ: *Dinic*:

נבנה רשת שכבתית מהרשת השיורית ע״י *BFS* מ-s. כל עוד t נגיש מ-s, נחפש מסלולים משפרים על הרשת השכבתית (כוללת רק קשתות משכבה לשכבה). כאשר t לא יהיה נגיש יותר מ-s, נזרוק את הרשת ונבנה רשת שיורית חדשה (וממנה שכבתית וכן הלאה, עד שאין מסלולים משפרים).
בכל שלב מציאת מסלול משפר, נזרים דרכו את הקיבול השיורי המינימלי ונמחק מהרשת השכבתית את הקשתות הרוויות (ונפחית קיבול מהקשתות האחרות במסלול ונתעלם מקשתות הפוכות לקשתות ברשת השכבתית, ששימוש בהן יוביל למסלול ארוך ממש מ-k).

נריץ אלגוריתם דומה ל-*DFS* על הרשת השכבתית החדשה מ-s עם שלוש הפרוצדורות:

- $Advance(u)$: אם אין קשת יוצאת מ-u, לך ל-$Retreat(u)$. אם יש קשת כזו (u,v) אז נמשיך ברקורסיה ($Advance(v)$). אם $v=t$ נבצע $Augment$.
- $Retreat(u)$: אם $u=s$ זהו סוף הטיפול ברשת השכבתית הנוכחית (סוף <u>פאזה</u>). אחרת נמחק את u ואת הקשתות הנכנסות אליו, נחזור לאב v שקרא ל-u ונבצע $Advance(v)$.
- $Augment$: זהו מסלול משפר, ולכן נמחק קשתות רוויות, נקטין קיבול שאר הקשתות במסלול ונחזור לצומת ההתחלה u של הקשת הרוויה הקרובה ביותר ל-s במסלול, ונבצע $Advance(u)$.

האלג׳ מוצא מסלולים משפרים ברשת השכבתית שהם מסלולים משפרים ברשת השיורית <u>באורך k</u>. לכן: אין מסלול ברשת השכבתית אמ״מ אין מסלול ברשת השיורית באורך k ולכן יש לעבור לפאזה הבאה. בפאזה החדשה: נעדכן את הזרימה ברשת המקורית שנוספה בפאזה הקודמת, ניצור רשת שיורית חדשה וממנה רשת שכבתית חדשה ונתחיל שוב את ה-*DFS* המשופר לעיל.

<u>נכונות</u>:

בכל פאזה הרשת השכבתית מכילה רק צמתים וקשתות שיכולים להופיע על מסלולים משפרים באורך k (מספר השכבות). לכן, בסיום הפאזה אין מסלולים משפרים ברשת השכבתית ולכן אין מסלולים משפרים מ-s ל-t ברשת השיורית באורך k!

<u>סיבוכיות</u>:

טענה: מרחק צומת מ-s עולה חלש (השכבה בה יימצא) בכל שיפור. לכן מספר הפאזות יהיה לכל היותר $|V|-1$: בסיום פאזה מרחק t מ-s ברשת השיורית גדול ממש מ-k ולכן כך יהיה גם ברשת השכבתית החדשה.

יעילות פאזה בודדת: נכנה קשת שמחקנו כקשת חסומה. בין שתי חסימות עוקבות מתבצעות $O(k)$ פעולות שזה $O(|V|)$. מספר החסימות לכל היותר $|E|$ ולכן סה״כ: $O(|E|\cdot|V|)$.

סה״כ: $O(|E|\cdot|V|^2)$ - שיפור לעומת *Edmonds-Korp*.

רשתות 0/1:

רשתות בהן כל קשת בעלת קיבול 1 (או 0). דיניץ על רשתות כאלה ייקח $O(|E|)$ לפאזה. בתחילת הדרך $G=G_f$ וכל הקיבולים הם 1 לאורך כל מהלך האלג׳: אם נמצא מסלול משפר, <u>כל</u> הקשתות עליו נהיות רוויות ולכן כולן יוצאות מהרשת השיורית וכל הקשתות ההפוכות נכנסות אליה עם קיבול שיורי 1. כל קשת מקורית או הפוכתה נמצאות תמיד ב-G_f עם קיבול 1, ובכל קשת הזרימה בכל שלב היא 0 או 1.

בביצוע *Augment* מוחקים / חוסמים k קשתות, עלות הפעולה $O(k)$. בין *Augment* אחד לשני מספר הצעדים פרופורציונלי ל-k+מספר הקשתות שנמחקו בגלל צמתים תקועים. לפיכך דיניץ רץ בזמן $O(|V|\cdot|E|)$ ברשתות אלו.

באופן כללי מספר הפאזות ברשת 0/1 הוא $O(\sqrt{|E|})$. כך נקבל:

- **במקרה הכללי**: $O(|E|^{\frac{3}{2}})$.
- **רשת מטיפוס 1**: רשת ללא קשתות מקבילות או אנטי-מקבילות (יכול להתקבל כאשר מפצלים קשת בעלת קיבול גדול מ-1 כדי להפוך רשת לרשת 0/1) $O(|E|\cdot|V|^{\frac{2}{3}})$, כי יש $|V|^{\frac{2}{3}}$ פאזות.
- **רשת מטיפוס 2**: רשת בה כל צומת בעל דרגת כניסה 1 או דרגת יציאה 1, $O(|E|\cdot|V|^{\frac{1}{2}})$, כי יש $|V|^{\frac{1}{2}}$ פאזות.

הערה: טיפוס הרשת נשמר גם ברשת השיורית.

<u>זיווג מקסימלי</u>:

G גרף דו צדדי, $X\uplus Y=V$, $E\subseteq X\times Y$, צריך למצוא אוסף מקסימלי של קשתות ב-E זרות בצמתים. האלג׳: מחברים s,t לכל הצמתים ב-X,Y בהתאמה, נותנים קיבול 1 לכל קשתות הגרף ובודקים זרימה מקסימלית. קשתות רוויות מהוות זיווג מקסימלי.

זרימה היא זיווג כי דרך כל קשת רוויה זורם 1. כיוון שהרשת מטיפוס 2 (דרגת כניסה/יציאה 1 לכולם פרט ל-s,t), אין 2 קשתות מאותו צומת ב-X או לאותו צומת ב-Y. עלות: $O(|E|\cdot|V|^{\frac{1}{2}})$ ע״י שימוש בדיניץ.

משפט Hall:

יהי G גרף דו"צ, $V = X \uplus Y$, $E \subseteq X \times Y$. לכל $A \subseteq X$ נסמן $\Gamma(A)$ את קבוצת השכנים ב-Y של כל הצמתים מ-A, ונניח $|X| = |Y|$. אז: ל-G קיים זיווג מושלם (זיווג בגודל $|X| = |Y|$) $\Leftrightarrow \forall A \subseteq X: |\Gamma(A)| \geq |A|$.

תרגילים וטענות (מהתרגול):

- בעיית הזיווג המקסימלי: בהינתן קבוצת בנים X, קבוצת בנות Y וקבוצת שידוכים אפשריים, מהו הזיווג המקסימלי האפשרי (זיווג שלם הוא $\min\{|X|,|Y|\}$ - זיווג בו כל בן משודך וכל בת משודכת).

פתרון: יוצרים גרף דו"צ ומוסיפים צמתים s,t: s יוחבר בקשתות עם קיבול 1 לכל X, t יוחבר בקשתות עם קיבול 1 לכל Y, הזיווגים האפשריים יהיו קשתות בין X ל-Y בעלי קיבול ∞. נמצא זרימה מקסימלית, והיא תתן את הזיווג.

נכונות: כל בן ב-X יכול לקבל זרימה 1 לכל היותר, ולכן יכול להעביר זרימה 1 לכל היותר. לכן לא יתכן מצב של זיווג בן אחד לשתי בנות. באופן דומה כל בת תזווג לבן יחיד כי יכולה להעביר ל-t זרימה לכל היותר בגודל 1. הרשת היא מטיפוס 2 ולכן סיבוכיות: $O(|E| \cdot |V|^{\frac{1}{2}})$.

- בהינתן גרף מכוון G, רוצים למצוא תת גרף שלכל הקודקודים דרגת הכניסה היא 1 ודרגת היציאה היא 1, או להראות שלא קיים כזה.

פתרון: יוצרים שני שכפולים ל-V: V_1, V_2. יוצרים מהם גרף דו"צ: לכל קשת מקורית (u,v) יוצרים קשת (u_1,v_2) בעלת קיבול בגודל 1. נמצא זיווג מקסימלי לקבוצת הקשתות שבנינו. אם מצאנו זיווג שלם אזי יש תת גרף – כך לכל קודקוד דרגת היציאה = דרגת הכניסה = 1.

- נתונים k מטוסים, n טייסות, m דיילים. רוצים למצוא שלשות (כאשר טייסות בוחרות מטוסים וגם דיילים בוחרים מטוסים) כך שמקסימום מטוסים ימריאו במקביל.

פתרון: מחברים s לכל הטייסות; מחברים את n הטייסות ל-k המטוסים לפי הנתונים; מחברים את k המטוסים לעותק נוסף של k המטוסים; מחברים את העותק הנוסף ל-m הדיילים לפי הנתונים; מחברים את הדיילים ל-t. נותנים לכל הקשתות קיבול 1 ובודקים זרימה מקסימלית. כך נקבל מספר מקסימלי של שלשות של טייסת,מטוס,דייל.

- נתונה רשת זרימה שערכה $|f| = 1000$. האם <u>בהכרח</u> יש ברשת זרימה שערכה $|f'| = 700$?

פתרון: כן. בונים מהרשת הישנה רשת חדשה שכל ההבדל הוא שיש קשת בעלת קיבול 700 מ-t אל t', שהוא הבור החדש ברשת החדשה. ה-*min-cut* ברשת החדשה הוא 700 כי ה-*min-cut* ברשת הישנה, השווה ל-*max-flow* באותה רשת, היה לפחות 1000. לכן ברשת החדשה יש זרימה מקס' של 700. כיוון שמ-t יוצאת קשת אחת עם זרימה 700 ל-t', משימור הזרימה נובע שחייב להיכנס ל-t 700. לפיכך ניתן להעביר ברשת המקורית 700 ל-t.

- נתונה רשת זרימה מכוונת עם קיבולים על הקשתות וזרימה מקסימלית f. לתאר אלג' הבודק האם קיימת ברשת קשת שהגדלת קיבולה תגדיל את הזרימה המקסימלית.

פתרון: קיים שיפור רק אם יש מסלול בזרימה המקסימלית בו יש קשת אחת בלבד רוויה (ושאר הקשתות על אותו מסלול לא). נעבור על כל הקשתות ונחזיק בהם שדה האם הן רוויות או לא. כעת נפתור ע"י *BFS* מ-s: אם נתקלים בקשת רוויה, ולא נתקלנו בכזו עד כה, נוסיפה למחסנית. אחרת נחזיר שלא קיים שיפור. המחסנית מתחזקת האם מסלול עד כה הוא רווי או לא.

- נתונה רשת זרימה בה כל הקשתות בעלות קיבולים זוגיים פרט לאחת e בעלת קיבול אי זוגי. נתונה זרימה מקסימלית בגודל אי זוגי. האם בהכרח e רוויה?

פתרון: כן. אם נניח בשלילה שלא, אז ניתן להקטין את קיבול אותה קשת לזוגי ועדיין לקבל את אותה זרימה מקסימלית. לפי *MCMF* גודל החתך המינימלי הוא גודל הזרימה המקסימלית. כיוון שכל החתכים כעת הם בגודל זוגי, בפרט גם המינימלי ולכן גם הזרימה המקסימלית – בסתירה.

- נתון גרף G ושני צמתים s,t. רוצים למצוא מס' מקסימלי של מסלולים זרים <u>בקשתות</u> מ-s ל-t.

פתרון: ניתן לכל קשת קיבול 1 ונמצא זרימה מקסימלית. נכונות: אם יש k מסלולים זרים, אז נזרים 1 על כל אחד מהמסלולים, ולא חרגנו מהקיבולים כיוון שהם מסלולים זרים. נקבל ש-$k \geq |f|$. כעת נניח באינדוקציה שעבור $|f| = l - 1$ מתקיים $|f| \geq k$. נוכיח נכונות ל-$|f| = l$: מוחקים את הקשתות שהזרימה בהן 0, ונשארים עם $|f|$. כיוון שהזרימה גדולה מאפס, קיים מסלול מ-s אל t. נבחר כזה ונמחק אותו ונקבל $|f| = l - 1$. מהנחת האינדוקציה יש $l - 1$ מסלולים זרים, ובתוספת המסלול שהורדנו יש בדיוק l. לכן $|f| = k$.

- **משפט מנגר לקשתות**: בגרף מכוון, s,t צמתים, המס' המקסימלי k של מסלולים זרים בקשתות שווה למס' המינימלי l של קשתות שיש למחוק כדי לנתק את s מ-t.

הוכחה: $l \geq k$ כיוון שחייבים למחוק לפחות קשת אחת מכל אחד מהמסלולים הזרים. $k \leq l$ כיוון ש: $|f|$ זרימה מקסימלית = k לפי תרגיל קודם. כמו כן לפי *MCMF* זה שווה לקיבול החתך המינימלי. כיוון שכל הקיבולים הם 1, זה בעצם שווה למס' המינימלי של קשתות (בחתך המינימלי). אם נמחק קשתות אלו ננתק את S מ-T ולכן את s מ-t. מכאן ש-$k = l$.

- נתון גרף לא מכוון G, רוצים לכוונו כך שלכל צומת תהיה דרגת יציאה לכל היותר 3, או להגיד שלא ניתן לבנות כזה.

פתרון: נבנה גרף דו"צ בו בצד אחד כל הצמתים מ-V, בצד שני צמתים חדשים e_i – צומת לכל $e = (u,v) \in E$. נחבר s לכל $v \in V$ עם קיבול 3, נחבר כל $u,v \in V$ אל e כך ש-$e = (u,v) \in E$ עם קיבול 1, ונחבר כל e ל-t עם קיבול 1. נמצא זרימה מקסימלית ונטען שקיים פתרון אמ"מ $|f| = |E|$. נניח שקיים פתרון כנדרש, אזי לכל $e = (u,v) \in E$ נכוון מ-u אל v ונזרים 1 על (u,e), ואם מ-v אל u: נזרים 1 על (v,e). נשלים את הזרימה מ-s אל t: $s \to u \to e \to t$. דרגת יציאה של כל קודקוד עד 3, ולכן כל קודקוד משתתף בעד 3 מסלולי זרימה כמתואר. לכן הזרימה על הקשתות (s,v) לכל היותר 3. כיוונו את כל הקשתות ולכן יזרים 1 על כל הקשתות (e,t) ונקבל $|f| = |E|$. כיוון שני באופן דומה.

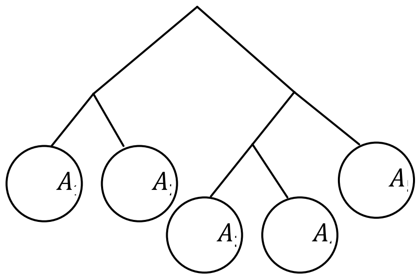
תכנות דינאמי:

תחליף לגישה רקורסיבית לפתרון בעיות, כאשר הפתרון הרקורסיבי יקר מדי כי הוא נתקל שוב ושוב באותן תתי בעיות ושוב ושוב פותר אותן. למשל, מספרי פיבונצ'י: הגיש הרקורסיבית מחשבת לכל n את כל מספרי הפיבונצ'י שקדמו לו, זמן ריצה אקספוננציאלי. הגישה האיטרטיבית לעומת זאת תבצע חישוב לינארי: כל צעד לחישוב F_k בדרך ל-F_n תחשב על בסיס החישובים הקודמים, כאשר מתחילים מ-F_0, F_1 ידועים.

כפל מטריצות יעיל:

נתונה סדרה של מטריצות $A_1, \dots, A_n$ לאו דווקא ריבועיות, כאשר כל מטריצה A_i היא מסדר $P_{i-1} \times P_i$ (מספר הטורים של A_i שווה למספר השורות של A_{i+1}. המכפלה היא מטריצה מסדר $P_0 \times P_n$. בגלל אסוציאטיביות של כפל מטריצות, ניתן לכפול לפי איזה סדר שנרצה, כאשר ברירת המחדל היא: $(((A_1 \cdot A_2) \cdot A_3) \dots)$.

המטרה: מציאת סדר הכפלה עם מספר מינימלי של פעולות כאשר מחיר כפל בודד של מטריצה $p \times q$ במטריצה $q \times r$ הוא pqr. עבור כל סדר הכפלה נייצג עץ בו העלים הם המטריצות. למשל: $(A_1 \cdot A_2)((A_3 \cdot A_4) \cdot A_5)$ ייוצג ע"י:



נסמן ב-$c[i,j]$ את המחיר המינימלי למכפלת המטריצות $A_i, A_{i+1}, \dots, A_j$, ומתקיים:

$c[i,j] = \min_{i \le k \le j}\{c[i,k] + c[k,j] + P_{i-1} \cdot P_k \cdot P_j\}$ – החלוקה שתתן עלות מינימלית המחושבת ע"י העלות המינימלית של כל חלק + עלות מכפלת שתי המטריצות שנקבל מחלוקה זו.

מקרה קצה: $c[i,i] = 0$ - זו מטריצה נתונה ואין מה לחשב.

עלות הביצוע הרקורסיבי $T[i,j]$ הוא:

$T[i,i] = 1$, ולכל $i \neq j$: $T[i,j] = \sum_{k=i}^{j-1}(T[i,k] + T[k,j] + 1)$ ונסמן זאת $T^*(j-i)$ – כפונקציה של הפער ולא האינדקסים ממש. בשיטה זו נקבל ש-$T^*(m) \sim 3^m$. בגישת התכנות הדינאמי:

יש $n + \binom{n}{2}$ תתי בעיות, כלומר $O(n^2)$, ונחשב כל אחת פעם אחת בלבד. בעיה עם הפרש $l = j - i$ תלויה בתתי בעיות עם הפרש $l >$. לכן נחשבן בסדר עולה החל מ-$l = 0$ ועד $l = n - 1$. תחילה נותנים ערך 0 לכל $c[i,i]$. לאחר מכן רצים על כל ההפרשים האפשריים (לכל הפרש l רצים על $c[i,i+l]$ לכל i), מאתחלים לערך ∞ ורצים בלולאה על k בה מבצעים את החישוב עצמו ועדכון הערך המינימלי ל-$c[i,j]$ ($j = i + l$). כמו כן מחזיקים שדה $\pi[i,j]$ שיחזיק את k עליו פיצלנו.

סיבוכיות: $O(n^3)$ במקום $O(3^n)$: לולאה על ההפרש l, הפרש j והפרש k.

- פתרון רקורסיבי: *top down*.
- פתרון דינאמי: *bottom up*.

חישוב תת סדרה משותפת מקסימלית:

תהיינה 2 סדרות $|X| = m, |Y| = n$: $\begin{matrix} X = A\boldsymbol{BCB}D\boldsymbol{AB} \\ Y = \boldsymbol{BDCAB}A \end{matrix}$ - תת הסדרה המקסימלית המשותפת היא המודגשת: *BCAB*.

נסתכל על האיברים האחרונים של הסדרות, X_m, Y_n. אם הם שווים, ברור שנשים אותם בתת הסדרה שנבנה. כך ניתן לפתור ברקורסיה בלי האיברים האחרונים. אם הם שונים, אז תת הסדרה המקסימלית תהיה או בין X_{m-1} ל-Y_n או בין X_m ל-Y_{n-1}. נסמן $X^{(i)}$ את הרישא ה-i של X.

תנאי העצירה הוא שאחת מהסדרות הגיע ל-$\emptyset$. מספר תתי הבעיות (כולל הסדרות הריקות) הוא $(m+1) \cdot (n+1)$.

שיטת התכנות הדינאמי:

נסמן ב-$LCS[i,j]$ את תת הסדרה המקסימלית בין $X^{(i)}, Y^{(j)}$. נכין מטריצה המחזיקה את ערכי *LCS*. האלגי המוצע הוא: במקום לחשב את *LCS* מפורשות, נחשב רק את האורך ומצביע לאיבר האחרון שלה, ומצביע למקום הקודם במטריצה בעזרתו נבנתה: אם $X_i = Y_j$ - באלכסון; אחרת או

למעלה במטריצה או שמאלה ($LCS[i-1,j]$ או $LCS[i,j-1]$). כך זמן הריצה הוא $O(m\cdot n)$, לעומת אם היינו שומרים בכל שלב את הסדרה במפורש, אז היתה העלות פי m או n יותר יקרה.

<u>דוגמא נוספת: פתרון משוואה</u>:

נתונים $a_1,\dots,a_n,b\in\mathbb{N}$ ו-$x_i\in\{0,1\}$, ונתונה המשוואה: $\sum_{i=1}^n a_i x_i = b$. לקבוע האם קיים פתרון, ולספור את מספר הפתרונות האפשריים.

פתרון זמן הריצה שלו תלוי ב-n ולא תלוי במספרים הנתונים יהיה אקספוננציאלי (בעיה NPC).

פתרון דינאמי:

נכניס פרמטר $M=\sum_{i=1}^n a_i$. אם יש פתרון אז $M\geq b$. נסתכל על תתי הבעיות $c[k,b]: \sum_{i=1}^k a_i x_i = b$, כאשר $b=1,\dots,M$, $k=1,\dots,n$.

פתרון הבעיה: $c[k,b]=\underbrace{c[k-1,b]}_{x_k=0}+\underbrace{c[k-1,b-a_k]}_{x_k=1}$, ו-$c[k,0]=1$ (הפתרון $x_i=0$ לכל i).

נקבל זמן ריצה $O(n\cdot M)$ לעומת $O(n\cdot 2^n)$ בפתרון הנאיבי.

<u>טריאנגולציה עם משקל מינימלי: *Minimum weight Triangulation*</u>:

נתון מצולע קמור P עם n קודקודים במישור ע"י העברת מיתרים שלא חוצים זה את זה. נפרק את P ל-$n-2$ משולשים, ונניח לכל משולש Δ יש משקל $w(\Delta)$, כאשר משקל הוא שטח המשולש. משקל הטריאנגולציה הוא סכום משקלי המשולשים, ורוצים למצוא טריאנג' בעלת משקל מינימלי.

נניח קודקודי המצולע הם $p_1,\dots,p_n$. נקח צלע כלשהי, למשל את $e=(p_n,p_1)$. למשולש שיושב על e יש קודקוד $1<k<n$ כך שמשולש זה מחלק את המצולע לשלושה מצולעים:

- P_1 עם k קודקודים.
- P_2 עם $n-k+1$ קודקודים.
- המשולש עצמו.

מתקיים: $w(P$ (טריאנגולציה מינימלית של $=\min_{2\leq k\leq n-1}\{w(p_1p_kp_n)+w_{min}(P_1)+w_{min}(P_2)\}$.

כל תת מצולע שמתקבל נפצל במיתר שממנו קיבלנו אותו (זה המשותף למשולש $p_1p_kp_n$), וכך יובטח שכל תת מצולע לאורך הרקורסיה מורכב מחלק רציף משפת P (המצולע המקורי) + מיתר. נתאר את P' ע"י (i,j) והכוונה ששפתו היא מקודקוד p_i עד קודקוד p_j נגד כיוון השעון.

ישנם $O(n^2)$ תת מצולעים.

אילוצים:

- $i\neq j$.
- אם $j=i+1$ אז המצולע הוא צלע בודדת ומשקלו 0.
- אם $j=i+2$ אז המצולע הוא משולש, ואין מה להיכנס ברקורסיה – נחזיר את משקלו.

נסמן באופן כללי: $W_{ij}=\min_{i+1\leq k\leq j-1}\{w(p_ip_kp_j)+W_{ik}+W_{kj}\}$. כעת נפתור את הבעיות לפי ההפרש $j-i$. עבור הפרש 1, $W_{i,i+1}=0$.

עבור הפרש $j=2\to n-1$: נרוץ על $i=1\to n-j$ ונגדיר: $W_{i,i+j}=\min\{w(p_ip_kp_j)+W_{ik}+W_{k,i+j}\}$ כאשר בין i ל-k ובין k ל-$i+j$ ההפרש קטן מ-j.

סיבוכיות: $O(n^3)$ (לולאות על i,j,k) לעומת אקספוננציאלית.

<u>תרגילים (מהתרגול)</u>:

- נתונה סדרה מספרים $x_1,\dots,x_n\in\mathbb{R}$, רוצים למצוא תת סדרה רציפה בעלת סכום מקסימלי.

פתרון: נחשב לכל i את: $S_i^+=$ סכום תת הסדרה הרציפה המקסימלי עד x_i (חייב להכיל אותו); $S_i^-=$ אותו סכום רק עד x_{i-1}. כעת חישוב האיבר הבא: $S_{i+1}^+=\max\{S_i^++x_{i+1},\ \underbrace{x_{i+1}}_{\text{מתחילים חדשה}}\}$, $S_{i+1}^-=\max\{S_i^+,S_i^-\}$. לבסוף ניקח את המקסימום מבין S_n^+,S_n^-. מציאת תת הסדרה ע"י הליכה מ-$i=n$ אחורה ומציאת i כך ש-$S=S_i^+$, ואז x_i מסיים את תת הסדרה. נמשיך כך אחורה לקבלת כל הסדרה. פתרון בזמן לינארי.

- **בעיית הסוכן הנוסע**: נתון גרף לא מכוון עם משקלות אי-שליליים. רוצים למצוא מעגל פשוט במשקל מינימלי העובר דרך כל הקודקודים.

פתרון: נניח שפתרון זה הוא OPT. נמצא מעגל שמשקלו לכל היותר $2OPT$. משתמשים בעובדה ש-$MST\leq OPT$ ולכן ניקח עפ"מ, לכל קשת נוסיף קשת אנטי-מקבילה כדי שנוכל לנוע במעגל (בכל קשת נשתמש פעם אחת בדיוק). כל קודקוד בדרגה זוגית ולכן קיים מעגל אויילר. הפתרון שלנו הוא בגודל $2MST\leq 2OPT$.

התאמת מחרוזות: ***String/Pattern Matching***:

הקלט הוא: מחרוזת טקסט T, מחרוזת תבנית P. רוצים לחפש איפה (אם בכלל) מופיעה P ב-T. נסמן ב-n את אורך T, וב-m את אורך P. נגיד ש-P מופיע במקום ה-i של T אם: $T_i = P_1, \dots, T_{i+m-1} = P_m$. הערה: מותרת חפיפה חלקית של התאמות.

פתרון נאיבי: לנסות את כל התזוזות האפשריות, לוקח $O(m \cdot n)$ זמן. להלן פתרון ב-$O(m+n)$:

נניח P,T מורכבות מ-$\{A,B\}$ בלבד. לאחר השוואה ראשונה, נבצע בהסתברות $\frac{1}{2}$ השוואה נוספת (לתו הבא), ובהסתברות $\frac{1}{4}$ השוואה נוספת (לתו אחריו) וכן הלאה. תוחלת מספר ההשוואות בתזוזה אחת: בערך 2.

אם בתזוזה מסויימת הצלחנו להתאים רישא של P שהוא $P[1,i]$, נחפש תזוזה מינימלית $0 <$ בה יש התלכדות בין רישא כלשהי של $P[1,i]$ לסיפא של אותה מחרוזת $P[1,i]$. למשל: $\begin{matrix} T = ABABABC \\ P = ABABC \end{matrix}$ - הרישא המתאימה היא $P[1{,}4] = ABAB$. במקרה זה AB היא רישא המתלכדת עם סיפא – לכן התזוזה הנדרשת לבדיקה הבאה היא 2 (ואז במקרה זה לאותה תזוזה גם תהיה התאמה).

נכין טבלה שלכל רישא $P[1,i]$ של P, נדע מה גודל הרישא המקסימלי של $P[1,i]$ שהוא גם סיפא של $P[1,i]$. נסתכל על i שהוא האינדקס המסיים את תת המחרוזת עליה מסתכלים, על $\pi(i)$ שהוא גודל אותו רישא של $P[1,i]$ שהוא גם סיפא, והתזוזה הנדרשת תהיה: $i - \pi[i]$.

לאחר חישוב π, נחזיק אינדקס q שיהיה: כמה תוים של p כבר מותאמים בהזזה הנוכחית.

נבצע השוואות ל-$P[q+1] \overset{?}{=} T[i]$, $P[q+2] \overset{?}{=} T[i+1]$... עד שנכשל או עד סוף P.

נניח k הוא אורך הרישא שהתאמנו, אז נקח את $\pi[k]$ מתוך הטבלה ונשים ב-q.

כדי לנצל את מירב האינפורמציה על כישלון בבדיקה מסויימת לקראת ההזזה והבדיקה הבאה, נחזיק טבלה $\delta[k,a]$ שתחזיק את גודל הרישא המקסימלית שהיא גם סיפא של $P[1,k]a$ (שרשור עם a, תו מהא״ב של P,T) לכל a בא״ב. π ניתנת לחישוב מתוך טבלה זו.

ניתן להסתכל על האלג׳ כעל אוטומט סופי:

אוטומט בעל $0,1,\dots,m$ מצבים. האוטומט קורא את T, וכאשר נמצא במצב k וקורא את a, עובר למצב $\delta[k,a]$, עד שמגיע למצב מקבל שהוא m. כשנגיע ל-m נודיע על הצלחה ונמשיך לרוץ עד סוף T. אופן פעולת האוטומט:

- כשהאוטומט במצב $q \equiv q$ התוים האחרונים שקראנו מ-T מתאימים ל-q התוים הראשונים שקראנו מ-P.
- מעבר δ: עוברים למצב k כאשר k הוא גודל הרישא המקסימלית של P שהיא גם סיפא של $P[1,q] \circ a$ (a התו הבא מתוך T כמו לעיל).
  אם $P_{q+1} = a$ אז $\delta[q,a] = q+1$ - יש לנו התאמה בתו נוסף ונתקדם. אם לא, נעבור לאותו k המתואר לעיל (חיפוש התאמה חדשה).

האלגוריתם של ***Knuth-Morris-Pratt (KMP)***:

דומה לרעיון האוטומט אך ללא תלות באות הקלט הבאה (a לעיל). נסתכל על רישא קטנה בתו אחד: במקום להסתכל על k שהוא הרישא המקסימלי המתאים לסיפא של $P[1,q] \circ a$, נסתכל על $k-1$: $P[1,k-1]$ שהוא סיפא של $P[1,q]$ (ללא a).

מעבר δ יוגדר שוב: נעבור למצב ה-k אם $P[1,k-1]$ הוא סיפא של $P[1,q]$. כמו פונקצית π שהוגדרה בהתחלה.

סיבוכיות:

חישוב π: $O(m)$. לולאת ה-*while* בחישוב π לא תתבצע יותר מ-$m-1$ פעמים, ולכן זמן החישוב הוא $O(m)$.

הפרוצדורה הראשית לוקחת $O(n)$ זמן. סה״כ: $O(m+n)$ ללא תלות בגודל הא״ב.

תרגילים (מהתרגול):

- נתונה מחרוזת T, רוצים למצוא את הרישא המקסימלית של T שהיא פלינדרום באופן יעיל.

פתרון: נרשום את $T = xy$ ואנו רוצים $x = x^R$ מקסימלי. נסתכל על: $T^R = y^R x$, ובעצם רוצים את רישא מקסימלית של T שהיא גם סיפא של T^R. נריץ *KMP* עם טקסט T^R ותבנית T. אם נסמן את אורך הרישא המקסימלית של P שהוא סיפא של $T[1,i]$ ב-$\sigma(i)$, אז האורך שרוצים הוא $\sigma(n)$. זמן הריצה: $O(m+n) = O(|T| + |T^R|) = O(2|T|) = O(|T|)$.

- תהי T באורך לפחות 10, רוצים למצוא $T = xyx$ חלוקה כך ש-$|y| \geq 10$, $|x|$ מקסימלי.

פתרון: בכל חלוקה כזו מתקיים: $|x| \leq \frac{n-10}{2}$. נריץ *KMP* עם $\left\lfloor\frac{n-10}{2}\right\rfloor$ התוים האחרונים של T כטקסט ו-$T\left[1, \left\lfloor\frac{n-10}{2}\right\rfloor\right]$ כתבנית. $\sigma\left(\left\lfloor\frac{n-10}{2}\right\rfloor\right)$ יתן את אורך ההתאמה המקסימלית.

- רוצים למצוא חלוקה $T = xyx$ כך ש-y היא תת מחרוזת של x (אם קיימת).

פתרון: מריצים *KMP* עם טקסט T ממקום $\left\lceil\frac{n}{2}\right\rceil + 1$ עד מקום n, עם התבנית $T[1,n]$ (כל הטקסט) כדי למצוע חלוקה $T = xyx$ עם x מקסימלי. לאחר מכן נריץ *KMP* עם x כטקסט ו-y כתבנית כדי לבדוק אם y תת מחרוזת של x.