אלגוריתמים – סיכומי הרצאות

13 בינואר 2011

מרצה: נתי ליניאל

מתרגל: צור לוריא

סוכם ע"י: אור שריר

פניות לתיקונים והערות: tnidtnid@gmail.com

אתר הסיכומים שלי: http://bit.ly/huji_notes

תוכן עניינים

חלק I

הרצאות

1 על הקורס

הקורס יתקיים ב־12 שבועות, כאשר ב־6 הראשונים יהיו בימי שני שעתיים של הרצאה וב־6 האחרונים רק שעה אחת.

ספר: קליינדרג וארדוש

2 אלגוריתמים חמדניים

הגדרה 2.1 אלגוריתם חמדן הוא אלגוריתם הפועל ע"פ העקרון הכללי "נסה לעשות את הצעד הנראה כרגע כמשתלם ביותר".

2.1 בעיות תזמון

דוגמא: יש לפנינו n משימות לביצוע, כאשר לכל משימה יש זמן התחלה a_i וזמן סימון b_i כאשר $b_i > a_i$. איננו יכולים לטפל בעת ובעונה אחת ביותר מאשר משימה אחת. המטרה היא לבצע תחת תנאים אלו, מספר מירבי של משימות.

פתרון:

1. מצא את המשימה T שזמן סיומה המוקדם ביותר, ז"א $b_i = \min\{b_j\}$.
2. סלק מכל הרשימה את כל הרשימות שמתנגשות בקטע $[a_i, b_i]$.

ברור שהאלגוריתם הזה מוצא קבוצה של משימות שביכולתינו לבצע כי הוא איננו מרשה ביצוע של שתי משימות החופפות בזמן. נטען שהמפתרון שמוצא האלגוריתם החמדן הוא הטוב ביותר במובן זב שאין שום קבוצה גדולה יותר מזו שהוא מצא שניתן לבצע.

טענה 2.2 (טענת עזר) יהיו $T_{i_1}, \ldots, T_{i_k}$ k המשימות הראשונות שמבצע האלגוריתם החמדן.
נניח שאם $T_{j_1}, \ldots, T_{j_k}$ היא סדרה של משימות שניתן לבצע (לא קיימות חפיפות במשימות), אזי $\max_{\alpha=1,\ldots,k} T_{j_\alpha} \geq b_{i_k}$.
כלומר נניח שהאלג' החמדן מבצע k משימות וגומר את האחרונה שבהן בזמן t. נניח שיש אוסף כלשהו אחר של k משימות שגם אותן ניתן לבצע, אזי זמן הסיום של המאחורים ביותר מבינהן הוא $\leq t$.

הוכחה: נוכיח באינדוקציה על k.

- אם $k = 1$ אז הטענה נובעת מהצעד הראשון של האלגוריתם.
- צעד האינדוקציה: ידוע לנו ש־$\geq b_{i_{k-1}}$ מהזמן המוקדם ביותר שבו ניתן להשלים $k-1$ משימות, ונניח בשלילה שיש k משימות שניתן להשלים מוקדם יותר (ממש) מאשר b_{i_k}.
  זוהי סתירה, כי המשימה ה־k שאותה מבצע "היריב" היא אפשרית גם לאלגוריתם החמדן, בגלל הנחת האינדוקציה, והיא גם עדיפה כי היא מסתיימת מוקדם יותר ולכן האלגוריתם היה בוחר בה, בסתירה לכך שקיימת סדרה שכזאת שהאלגוריתם לא בחר.

■

טענה 2.3 האלגוריתם הנ"ל מוצא את הפתרון האופטימלי.

הוכחה: נניח שהחמדן ביצע k משימות ויש יריב שביצע m משימות. נרצה לטעון ש־$m \leq k$.
אם $m > k$, נביט בזמן שבו השלים היריב את משימתו ה־k, ע"פ טענת העזר זהו זמן מאוחר יותר מזמן הסוף של האלגוריתמם החמדן, ולכן האלגוריתם החמדן יכול היה עדיין לבצע גם את המשימה ה־$k+1$ של היריב ־ סתירה! ■

הערה 2.4 נסתכל על גרף בן הקודקודים של V הם הקטעים (המשימות) ובין שני קדוקדים יש צלע אם"ם המשימות נחתכות. מחפשים אנטי קליקה הגדולה ביותר בגרף הזה, ז"א קבוצה גדולה ביותר של קודקודים שאף שניים מהם אינם שכנים. למרות שבאופן כללי הבעיה של מציאת אנטי קליקה גדולה ביותר היא קשה (NP קשה) אבל במקרה זה, מדובר במקרה מיוחד בו הגרף הוא גרף אינטרוואלי.

בעיה: בבי"ס יש שיעורים שונים שצריך ללמד, כאשר לכל שיעור יש זמן התחלה וזמן סיום. מהו המספר המזערי של הכיתות שיספיקו על מנת שניתן יהיה ללמד את כל השיעורים.

פתרון: נתרגם תחילה לבעיה של גרפים, כאשר כל שיעור הוא קטע בזמן, ולכל קטע (השיעור) נתאים קדקוד, שני קדקודים שכנים אם זמניהם בחפיפה. את השיוך לכיתות ניתן לקודד ע"י מתן צבעים לקודקודים, כך ששני קודקודים שווי צבע חייבים להיות בלתי שכנים.
בהינתן הגרף, המטרה היא למצוא צביעה של קודקודיו במספר מזערי של צבעים כך שלכל שני קדקודים שהינם צבועים בצבעים שונים.
הבעיה של מציאת מספר צבעים היא באופן כללי קשה NP, אבל כיוון שאני מתעסקים בגרפים אינטרוולים, למקרה פרטי זה, יש פתרון ע"י אלגוריתם חמדן.
נזכיר שקליקה בגרף, זו קבוצת קדקודים שכל שנייה מהם מחבורים בצלע. קל לראות שאם בגרף נתון G יש קליקה בת k קודקודים, אז יידרשו לפחות k צבעים ע"מ לצבוע את G.
כפי שנראה, בגרף אינטרוולים, מספר הצבעים שווה הגודל המירבי של הקליקה. נשים לב שמספר הצבעים הוא המס' המזערי של כיתות שיספיקו, והגודל המירבי של הקליקה הוא המספר המירבי של שיעורים המתקיימים סימולטנית (באותו הרגע).

טענה 2.5 נניח שנתון לנו אוסף של שיעורים (כ"א הוא קטע בזמן) נניח שהמספר המירבי של שיעורים המקיימום בו זמנית הם d, אז נחוצות אבל גם מספיקות d גיתות ע"מ לקיים את ההוראה. יתר על כן, האלגוריתם החמדן שנתאר להלן מצא דרך למקם את השיעורים בכיתות:

1. עבור על השיעורים ע"פ סדר עולה של ריגעי ההתחלה.
2. שכן את השיעור הבא בכיתה הפנוייה הראשונה ־ כלומר יש להראות שתמיד תהיה כיתה פנוייה בה ניתן למקם את השיעור שבו מטפלים כרגע.

הוכחה: נוכיח את השלבים:

1. אופן הסידור מגדיר שלא נמקם כיתות בחפיפה
2. נניח בשלילה שיש רגע מסויים שבו אנו נכשלים, אבל זה בסתירה לכך שהקצאנו מספר מקומות כמספר המקסימלי של שיעורים חופפים.

■

2.2 בעיית זיכרון מטמון

בזיכרונות של מחשבים מודרניים יש היררכיה שבה מהירות ומחיר מאזנים זה את זה. באופן אופייני הזיכרון במחשב מאורגן במספר רמות, ככול שהרמה גבוהה יותר, הגישה לזיכרון מהירה יותר, מאידך גיסה, זכרונות אלו הם יקרים ולכן בד"כ משתמשים בזכרונות זולים אך איטיים יותר.

נדבר על מצב שבו יש למחשב שלנו זיכרון מהיר הנקרא זיכרון מטמון (Cache) וזיכרון איטי גדול בהרבה. בזיכרון המטמון יש k יחידות זיכרון, ובחיצוני $n \gg k$. יש גישות לזיכרון שבהן אנו נזקקים ל"דף" המצוי כרגע בזיכרון המטמון ("פגיעה" - hit), אבל לפעמים הדף הרצוי אינו שם ויש להביאו לזיכרון המהיר מן הזיכרון החיצוני ("החטאה" - miss). הפעולה של החלפה בין דף שבזכרון המטמון לבין דף בזיכרון החיצוני היא יקרה (ולפעמים מהווה צוור הבקבוק של הבעיה), ואנו רוצים כמובן למעט בהחטאות. במקרה של החטאה יש לסלק (evict) דף כלשהו מן הזיכרון מהמיר ע"מ לשכן במקומו את הדף הרצוי מהזכרון האיטי. תיאור מלא של אלגוריתם לבעיה זו הוא כלל המגדיר איזה דף לסלק במצב נתון של החטאה.

(החלק הבא לא בחומר ומיועד להעשרה) הדבר הרצוי הוא אלגוריתם שיצבע מספר קטן של החטאות יחסית למה שניתן היה באופן אופטימלי. ז"א נשווה את מספר ההחטאות של האלגוריתם הנתון עם מספר ההחטאות שהיה עושה "האלגוריתם הנביא" היודע מראש מה תהיה סדרת הבקשות. הגישה הזאת נקראת חישוב מקוון - Online. מנתחים בגישה זו אלגוריתם LRU - Least Recently Used.

(חזרה לחומר) בעיה: נתון זיכרון מטמון בגודל k שמאותחל לקבוצת דפים כלשהי, נתונה רשימה סדורה של דרישות של הדפים שיידרשו.

המטרה: אלגוריתם לטיפול בהחטאות כך שהיה לו מספר מזערי של החטאות לסדרת הבקשות הנתונה.

האלגוריתם: האלגוריתם (החמדן) הנקרא FF - Farthest into the Future, כלומר בהינתן רשימת הבקשות אנו נסלק בזיכרון המטמון את הדף שבקשת הגישה הראשון אליו היא בעתיד הרחוק ביותר.

משפט 2.6 לאלגוריתם ה-FF יש לכל סדרת בקשות, המספר המזערי האפשרי של החטאות.

הוכחה: הרעיון בבסיס ההוכחה הוא שבהינתן קלט ו-S הוא אלגוריתם אופטימלי לקלט זה (אבל לאו דווקא לקלטים אחרים). נראה שבהינתן S, ניתן למצוא אלגוריתם אחר S' שמחירו (לקלט הנתון) אינו עולה על זה של S אבל התנהגותו של S' מתלכדת עם זו של FF למשך זמן ארוך יותר מאשר ל-S, למעשה נקבל כך את אי השוויון:

$$cost\,(S) \geq cos\,(S') \geq \ldots \geq cost\,(FF)$$

אומרים שאלגוריתם לבעיה זו הוא מצומצם אם הוא מוציא איזשהו דף מן הזכרון המהיר רק ע"מ להכניס במקומו דף הנדרש כרגע ואינו בזיכרון המהיר.

טענה: יש אלגוריתם אופטימלי מצומצם.

הוכחה: נראה שלכל אלגוריתם לבעיה A, יש אלגוריתם מצומצם A' שמחירו אינו גבוה יותר.
נניח שברגע כלשהו A מוציא מזכרון המטמון דף x ומכניס במקומו דף אחר y שאינו נדרש כרגע, איך נהג A'? הוא יוותר על הפעולה הזו ואז בעתיד יידרש הדף y אז הוא יכניס אותו.

חזרה להוכחה הראשית. בהינתן אלגוריתם S, נרצה לבנות אלגוריתם S' המתלכד עם FF לעוד צעד אחד נוסף (מעבר למה שעשה S) כך ש-$cost\,(S') \leq cost\,(S)$.
נדרש איזשהו דף מהזיכרון החיצוני, FF סילוק למענו איזשהו דף e ו-S סילוק למענו את F ו-S' סילק גם הוא את e. (תמונה)
מגיעה דרישה ל-x מהחיצוני, FF מסלק את e ו-S מסלק את f.
תיאורו של S' > S' מכאן ואילך S' יפעל בדיוק כמו S כל עוד הדבר אפשרי.
מתוך ההגדרה של אלגוריתם FF נובע מהדרישה ל-f תבוא לפני הדרישה ל-e.
באילו מצבים יהיה קושי ל-S' לנהוג בדיוק כמו S?

1. S מכניס דף כלשהו g במקום e במקרה זה. מכאן ואילך S, S' זהים ומחירם זווה, ו-S' נהג כמו FF צעד אחד יותר.

2. כשמתבקש להכניס את e או את f. כפי שראינו הדרישה ל־f תבוא קודם. אנו מטפלים אם כך במצב שבו בתעוררה לראשונה הדרישה ל־f.

(א) אפשרות אחת: S מחליט להוציא את e ע"מ להכניס את f. במקרה הזה S' נמנע מכל פעולה, ומנקודה זו ואילך שאר פעולותיהם שווים ובעצם $cost(S') = cost(S) - 1$.

(ב) אפשרות שנייה: S מכניס את f במקומו של דף אחר e'. במקרה זה S' יחליף את e' ב־e.

מכאן ואילך ל־S ול־S' יש אותו הזיכרון ולכן S' יכול לחקות את S באופן מושלם. כמובן S' אינו מצומצם, אבל ע"פ טענת עזר קודמת, ניתן להחליפו באלגוריתם אחר S'' מצומצם במחיר לא גבוה יותר. ■

2.3 דחיסת מידע ־ קידוד הופמן

הגדרה 2.7 דחיסת מידע ־ רוצים לתאר בצורה קצרה ככל האפשר את המידע שיוצר המשדר, בכפוף להנחה שבמקלט ניתן לפתח נכונה את התשדורת.

קונקרטית: המשדר יוצר תשדורת בגודל של n סמלים, הסמל ה־i מופיע בשכיחות p_i, כאשר $p_1, \ldots, p_k \geq 0$ ו־$\sum p_i = 0$.
רוצים לקודד כ"א מהסמלים בא"ב של השמדר במחרוזות של ביטים כך ש־(א) האורך המתקבל יהיה קצר ו־(ב) התשדורת תהיה ניתנת לפיענוח יחיד.
תנאי מספיק לכך שניתן יהיה לפענח ביחידות בודעות כשהן מקודדות בביטים היא תנאי העדר הרישא (prefix-free).
לכל אות בא"ב של המשדר מתאימים למחרוזות ביטים ואף אחת מהמילים איננה רישא של האחרת.

טענה 2.8 אם נקודד כל אות בא"ב של המשדר במחרוזות של ביטים ואם אף מחרוזת כזו איננה רישא של אחרת, אז כל שרשור של מחרוזות כנ"ל ניתן לפיענוח יחיד.

α היא רישא של β אם"ם α היא אב קדמון של β בעץ הבינרי. אוסף המילים המקיים את תנאי העדר הרישא הוא "חזית" בעץ, ז"א זהו אוסף העלים של איזהשהו תת עץ סופי.

השאלה: למצוא עץ בינרי עד n עלים $x_1, \ldots, x_n$ כך שהאורך הממוצע של תשדורת יהיה קצר ככל האפשר.
כלומר למזער את הסכום $\sum p_i h_i$ כש־h_i הוא העומק של העלה ה־i בעץ = אורך הקידוד של הסמל ה־i.

שאלה: מהן התכונות של הקידוד שיבטיחו שמהקליט יוכל לפענח באופן חד משמעי את ההודעה המתקבלת?

תשובה: ראינו ששאר הקידוד שלו הוא חסר רישא ־ ז"א לכל $i \neq j$ מתקיים שהמחרוזת המקודדת את האות i איננה רישא של המחרוזת המקודדת את $j \Leftarrow$ הפענוח הוא יחיד.
גם ראינו שיש התאמה חח"ע בין מילותיו של קוד חסר רישא לקבוצת העלים של עץ בינרי.

הבעיה: (ניסוח מחדש) נתונים $p_1, \ldots, p_n \geq 0$ כך ש־$\sum p_i = 1$. מצאו עץ בינרי בעל n עלים בעומקים $l_1, \ldots, l_n$ כך ש־$\sum p_i l_i$ מינימלי.

הערה 2.9 $\sum p_i l_i$ הוא האורך הממוצע של (=תוחלת האורך) קידוד של אות בתשדורת המקום.

האלגוריתם: (קידוד הופמן) מצא את שני הסמלים הנדירים ביותר (ש־p_i שלהם מינימלי), נאמר x, y והחלף אותם באות אחת חדשה z (שאינה בא"ב המקורי) שההסתברות שלה $p_z = p_x + p_y$ וחזור על התהליך עבור הא"ב המצומצם.

נכונות: מדוע קידוד הופמן נותן קידוד אופטימלי?

1. נשים לב שבעץ אופטימלי לכל עלה יש אח (כלומר עץ שלם), אחרת במידה לקודקוד יש בן עלה בודד, ניתן למחוק את הקודקוד ולשים במקומו את העלה.

2. נביט בעץ האופטימלי: אם יש בו שני עלים בעומק שונה אז בעלה העמוק יותר יושבת האות בעלת ה־p הקטן ביותר.
אם $l_\alpha < l_\beta$ ו־$p_\alpha < p_\beta$ אז עדיף לנו להחליף בין הקודקודים כדי שנייצג את האות השכיחה יותר בפחות סימנים.

3. אם יש שני עלים בעץ באותו העומק, אז הערך של $\sum p_i l_i$ לא ישתנה ע"י החלפת האותיות שבעלים אלה. לכן נובע ששני העלים בעלי המשקל הנמוך ביותר הם באותו העומק, וזהו העומק המירבי.
4. בזכות (3) ו־(1) ניתן להעביר את שתי האותיות הקלות ביותר להיות אחיות בעץ ללא שינוי ב־$\sum p_i l_i$.
5. השינוי במחיר של העץ ע"י החלפת שני האותיות הקלות ביותר באות z השווה לסכומם, הוא הפחתה ב־p_x+p_y ולכן הפתרון הטוב ביותר מתקבל ע"י פתרון הבעיה בא"ב המצומצם (שבו השמטנו את x, y והוספנו את z ו־$p_z = p_x + p_y$) ועוד תוספת של $p_x + p_y$.

שאלה: איך מודדים כמות אינפורמציה? אנטרופיה.

דוגמא: נסתכל על מקור בו $p_0 = \frac{1}{2}, p_1 = \frac{1}{2}$, לעומת מקור בו $p_0 = \frac{1}{3}, p_1 = \frac{2}{3}$. המקור הראשון מייצר יותר מידע כיוון שיש לו יותר כוח הבעה.

הגדרה 2.10 יהי $p_1, \ldots, p_k \geq 0$ ו־$\sum p_i = 1$ אז אנטרופיה מוגדרת ע"י

$$H(p_1, \ldots, p_n) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} = -\sum p_i \log_2 p_i$$

יש לנו מקור ונרצה לכמת את קצב יצירת האינפורמציה שלו. נבחר N גדול מאוד $(N \to \infty)$ ונשאל כמה תשדורות A_N שונות באורך N הוא יכול לייצר. האינפורמציה היא $\log_2 A_n$.
אם יש אוסף נוסף בעל הסתברות קטנה מאוד של תשדורות אפשרויות (אבל מאוד נדירות) נוכל להתעלם מהן.
נקבע $\epsilon > 0$ ונשאל מהו מספר התשדורות השכיחות ביותר שהסתברותן הכוללת היא $1 - \epsilon <$, ונקרא למספר זה $B_n(\epsilon)$.
נראה אם כן ש־$\log_2 B_n$ הוא המדד המתאים לכימות קצב יצירת האינפורמציה של המקור. קצב יצירת האינפורמציה הוא $\frac{1}{N}\log_2 B_n$.

טענה 2.11 לפי הגדרה זו, אם יש לנו מקור המשדר בא"ב של k אותיות עם שכיחויות הופעה $p_1, \ldots, p_k \geq 0$ ו־$\sum p_i = 1$, אז קצב ייצור האינפורמציה של מקור זה הוא

$$H(p_1, \ldots, p_k) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

רעיון ההוכחה: יש משפט יסוד בהסתברות שנקרא "חוק המספרים הגדולים", לפי משפט זה כשהמקור הנ"ל ישדר N פעמים, אז מספר המופיעים של האות i יהיה "בערך" $p_i N$.
לכן אם נתרכז רק במילים באורך N שבהן לכל i האות ה־i מופיעה "בערך" $p_i N$ פעמים נקבל כמעט (במובן ההסתברותי) את כל התשדורות מאורך N.
מספר המילים בא"ב $1, \ldots, k$ שבהן האות ה־i מופיעה הוא בקירוב $p_i N$ ולכן

$$B_N = \frac{N!}{(p_1 N)!\,(p_2 N)! \cdots (p_k N)!}$$

2.3.1 ייצוג כבעיית אופטימיזציה

יש איזושהי קבוצה D ופונקציה ממשית $f:D\to\mathbb{R}$ ומחפשים $\sup_{x\in D} f\left(x\right)$.

הגדרה 2.12 אופטימליות מקומית - $x\in D$ הוא המקסימום אם"ם לכל $y\in D$ $f\left(x\right)\geq f\left(y\right)$.
ע"מ ש־$x\in D$ תהיה נקודת המקסימום, הכרחי ש־$f\left(x\right)\geq f\left(y\right)$ לכל y ש־"קרוב" ל־x.

נניח שידוע לנו מהו העץ האופטימלי בבעיית קידוד המקור. נאמר שזהו עץ עם עלים בעומקים $l_1\geq\ldots\geq k_k$.

בעיה: כיצד נתאים את n אותיות הא"ב לעלים ע"מ שהמרחק המשוקלל יהיה מזערי. מהי התמור π כך ש־$\sum l_i p_{\pi(i)}$ מינימלי?

הערה 2.13 נשים לב כי זו בעייה כללית. כאשר נתונים $a_1\geq\ldots\geq a_k\geq 0$ וגם $b_1\geq\ldots\geq b_k\geq 0$ אז מהי התמורה π (ההתמאמה בין הקבוצות) כך ש־$\sum a_i b_{\pi(i)}$ מקסימלי או מינימלי?

לא ברור בדיוק למה החלק הבא:

$$\left(a_i b_r + a_j b_s\right)-\left(a_1 b_s + a_j b_r\right)=\left(a_i - a_j\right)\left(b_r - b_s\right)$$

2.4 מטרואידים

יש קבוצת איברים E סופית ומשפחה של תת קבוצות של Eו למשפחה שנקרא F. יש פונקציה $w:E\to\mathbb{R}^+$.

בעיית האופטימיזציה: מצא את $A\in F$ שבשבילה $\max\sum_{x\in A} w\left(x\right)$ (בדומה להסבר בקודי הופמן).

דוגמא: נסתכל על קבוצת עובדים וקבוצת משימות, ובין עובד ומשימה קיימת צלע עם משקל w_{ij} המייצג את מידע המיומנות של עובד i עבור משימה j.
המטרה היא להשים עובדים למשימות כך שיושג רווח מירבי (מבחינת ניצול כוח העבודה).
במקרה זה E היא קבוצת הצלעותשל הגרף הנ"ל, ω הוא המשקל של הצלעות, ו־F הוא אוסף הזיווגים בגרף.
זיווג בגרף זהו אוסף של צלעות שזרות זו לזו (זאת אומרת אף קודקוד אינו מכוסה פעמים).
נשים לב שבמצב זה האלגוריתם החמדן לא נותן את הפתרון הטוב ביותר! (תמונה בטלפון).

הערה 2.14 למדנו שהאלגוריתם החמדן פותר את בעיית העץ הפורש המינימלי.
בעיית העץ הפורש המינימלי שקולה לבעיית העץ הפורש המקסימלי.

- הקלט: $G=\left(V,E\right)$ בגרף קשור $W:E\to\mathbb{R}^+$. מהו העץ הפורש המינימלי שבשבילו $\sum\omega\left(e\right)$ מזערי.
- נניח שהמספר M גדול מכל $\omega\left(e\right)$ (לכל $e\in E$) ואז $w'\left(e\right)=M-w\left(e\right)$.
- טענה: אם T עץ כלשהו, אזי $\sum_{e\in T} w'\left(e\right)=\left(n-1\right)M-\sum w\left(e\right)$.
  פתרון בעיית min ST שקולה לבעיית max ST.
- תיאור של בעיית max span tree במסגרת שניסחנו היא: E קבוצת הצלעות של הגרף, w פונקציית משקל, ו־F הוא יער.

תזכורת: נוסחת Stirling לעצר היא $n!=\left(1+o\left(1\right)\right)\left(\frac{n}{e}\right)^e\sqrt{2\pi n}$

2.4.1 בעיית היער הפורש המקסימלי

יש קבוצה סופי ("קבוצת בסיס") E ומשפחה F של תתי קבוצות של E (ז"א אם $A \in F$ אז $A \subseteq E$). בהינתן פונקציה $W : E \to \mathbb{R}^+$, נרצה למצוא $\max_{A \in Y} W(A)$ כאשר $W(A) = \sum_{x \in A} W(x)$.

הנחה: המשפחה F היא תורשתית, כלומר אם $A \in F$ ו־$B \subseteq A$ אז גם $B \in F$.

ראינו שאם E קבוצת הצלעות של גרף ו־F הוא כל ה־$A \subseteq E$ כך שהן זיווג, אז במקרה זה האלגוריתם החמדן לא בהכרח פותר את הבעיה.

האלגוריתם החמדן: (שלא תמיד פותר את הבעיה)

1. צא מ־$A = \emptyset$.
2. מדי צעד צרף ל־A איבר $x \notin A$ כך שגם $A \bigcup \{x\} \in F$ וכך ש־$W(x)$ מירבי.

מצד שני, למדנו שאם F הוא אוסף היערות בגרף כלשהו אז האלגוריתם החמדן דווקא כן מובטח שיצליח וימצא את האופטימום.

הגדרה 2.15 יהיו שתי קבוצות $A, B \in F$ כך ש־ $|A| > |B|$ ונרצה לדעת אם קיים $x \in B \backslash A$ כך ש־$A \bigcup \{x\} \in F$.
אם לכל A, B כנ"ל יש x כנ"ל אומרים שהמשפחה F מקיימת את תנאי ההחלפה.

הערה 2.16 נשים לב שאם F הוא אוסף הזיווגים בגרף, אז F לא בהכרח מקיימת את תנאי ההחלפה.

משפט 2.17 תהיה F משפחה תורשתית של תת קבוצות של הקבוצה הסופית E, אז האלגוריתם החמדן פותר את הבעיה $\max_{A \in F} W(A)$ לכל פונקציית משקל W אם"ם F מקיימת את תכונת ההחלפה.

מסקנה 2.18 מסקנות מהמשפט:

1. אם המשפחה F מקיימת את תנאי ההחלפה, אז לכל $W : E \to \mathbb{R}^+$, האלגוריתם החמדן ימצא $A \in F$ כך ש־$W(A)$ מקסימלי.
2. אם F איננה מקיימת את תנאי ההחלפה, אז יש $W : E \to \mathbb{R}^+$ שתכשיל את האלגוריתם החמדן ־ ז"א הקבוצה שאותה ימצא האלגוריתם החמדן, היא איננה ממקסמת את $W(A)$ על פני $A \in F$.

הוכחה: (2) תהיה F משפחה תורשתית של תת קבוצות של E, ויהיו $A, B \in F$ המפרות את תנאי ההחלפה, כלומר $|A| > |B|$ אבל לכל $x \in B \backslash A$ אז $A \bigcup \{x\} \notin F$.
נגדיר ש־$W(a) = 1 + \epsilon$ לכל $a \in A$, ונגדיר ש־$W(x) = 0$ לכל $x \notin (A \bigcup B)$. לכל איבר $b \in B \backslash A$ נגדיר $W(b) = 1$.
האלגוריתם החמדן יצרף בזה אחר זה את איברי A ויגיע למשקל כולל $W(A) = (1 + \epsilon)|A|$.
בנקודה זו האלגוריתם אינו יכול לצרף עוד איברים מ־$B \backslash A$ בגלל הפרת תנאי ההחלפה. אולי הוא יכול לצרף מ־$E \backslash (A \bigcup B)$ אבל אלא משקלם אפס.
אפשרות אחרת הוא לבחור ב־B $W(B) \geq |B|$ והוא נכשל אם $|B| > (1 + \epsilon)|A|$.
כיוון שע"פ ההנחה $|A| > |B|$ אז קיים $\epsilon > 0$ שבשבילו אי השוויון לעיל מתקיים, ולכן בשבילו האלגוריתם החמדן נכשל.. ∎

הגדרה 2.19 משפחה תורשתית של קבוצות סופיות המקיימת את תנאי ההחלפה נקראת אוסף הקבוצות הבלתי תלויות של מטרואיד.

דוגמאות:

1. E היא קבוצה סופית של וקטורים במרחב וקטורי כלשהו. $E \supseteq A \in F$ אם"ם A היא בלתי תלויה לינארית.
F היא תורשתית כי כל תת קבוצה של קבוצה בת"ל היא בת"ל.
F מקיימת את תנאי ההחלפה. נניח $A, B \in F$ ושתיהן בת"ל אך $|A| < |B|$ ולכן $\dim(\mathrm{Span}(B)) > \dim(\mathrm{Span}(A))$ ולכן קיים $x \in B$ שאיננו שייך למרחב הנפרש של A, ולכן אם נצרף אותו ל־A אז הקבוצה החדשה תישאר בת"ל.

2. E היא קבוצת הצלעות של גרף סופי $G = (V, E)$ ו־$A \in F$ אם $A \subseteq E$ ו־A מגדיר יער (ז"א הגרף (V, A) הוא יער ־ גרף חסר מעגלים).
תת גרף של יער הוא יער ולכן F תורשתית.
תכונת ההחלפה מתקיימת. יהיו $A, B \subseteq B$ כך ש־$(V, A), (V, B)$ הם יערות ו־$|A| < |B|$.
את הצלע $y \notin A$ ניתן לצרף ל־A בלי להפר את תכונת היער אם"ם y מחברת בין שני רכיבי קשירות שונים של A.
ע"מ להוכיח את כלל ההחלפה, יש להראות את הדבר הבא:
יהיו $(V, B), (VA)$ שני יערות על אותה קבוצה קודקודים V ונניח ש־$|A| < |B|$ אז יש צלע $x \in B \backslash A$ המחבר בין שני רכיבי קשירות שונים של A.
נסמן ב־$a_i \geq 1$ $(i = 1, \ldots, k)$ את מספר הקודקודים ברכיב ה־i של A. נאמר שצלע מ־B פסולה אם שני קודקודיה הם באותו רכיב קשירות של A.
נרצה להוכיח שלא כל הצלעות ש־B פסולות (צלע ב־B שאינה פסול מקיימת את הנדרש).
הרכיב שגודלו a_i פוסל לכל להיות $a_i - 1$ צלעות של B (כי על קבוצת קודקודים בעלת t קודקודים יש ליער B לכל היותר $t - 1$ צלעות).
יוצא שנפסלו לכל היותר $A = \sum (a_i - 1)$ צלעות ב־B, ולכן קיימת לפחות $|B| - |A|$ שאינן פסולות, וכל צלע כזו מקיימת את התנאי.

הערה 2.20 כשמפתחים את תורת המטרואידים במלואה, מגדירים מלבד קבוצות בלתי תלויות עוד מושגים חשובים:

- בסיס ־ קבוצה בת"ל מגודל מירבי (מתאים לעץ פורש).
- מעגל ־ קבוצה תלויה מינימלית.
- חתך ־ קבוצה מינימלית של איברים הפוגשת כל בסיס.

3 תכנון דינאמי (Dynamic Programming)

בעיות המתאימון לפתרון בשיטה זו מאופיינות ע"י כך שיש מושג של תת בעיה ואם פתרון הבעיה שלפנינו כולל בתוכו איזושהי תתי בעייה שאלה אנו פותרים, אז בה"כ כם את התת בעיה אנו נופתור בצורה אופטימלית. הפתרון האופטימלי של הבעיה כולה מתקבל מצירוף של פתרונות אופטימלים לתת בעיות.

דוגמא: (נדגים את הרעיון ע"י דוגמא זו) נתונות שתי נקודות x ו־y וקודקוד נוסף z, וקיימות n צלעות בין x ל־z ו־m בין z ל־y, ונרצה למצוא את המסלול הקצר ביותר בין x ל־z.
קל לראות שאם נרצה לעבור על כל המסלולים אז נצטרך לעבור על nm אפשרויות, אבל נשים לב כי מתקיים כי $d(x, y) = d(x, z) + d(z, y)$ ולכן אם נמצא את הפתרון האופטימלי ל־$d(x, z)$ ו־$d(z, y)$ אז נוכל למצוא את הפתרון המינימלי ל־$d(x, y)$ ולכן נעבור רק על $n + m$ אפשרויות.

3.1 כפל מטריצות

נזכיר: אם $A_{r \times s}$ ו־$B_{s \times t}$ מטריצות אז $AB = C_{r \times t}$. בנוסף נזכיר כי כפל מטריצות הוא אסוציאטיבי.

הקדמה: אם A ו־B כנ"ל, אז ההכפלה הנאיבית של A ו־B עולה לנו $O\left(rst\right)$ פעולות אריתמטיות אלמנטריות.
נתונה לנו סידרה של מטריצות $A_1, \ldots, A_n$ כך שהמטריצה ה־A_i היא במימדים $\alpha_{i-1} \times \alpha_i$ ולכן הכפל מוגדר.
בגלל האסוציאטיביות המכפלה הזו מוגדר היטב והיא מטריצה $\alpha_0 \times \alpha_n$ ותוצאה החישבו אינה תלויה באופן שבו נמסגר את הביטוי.
מצד שני המחיר החישובי תלוי באופן המיסגור הנבחר.

בעיה: איך נמצא את המסלול הזול ביותר מהבחינה החישובית?

פתרון 1: נעבור על כל אופני המסגור השונים, נחשב לכ"א מהם את המחיר החישובי ונבחר באופציה היעילה ביותר.
הרעיון הזה לא מוצלח מפני שמספר המיסגורים הוא מספר קטלן הגדל מעריכית עם n.

פתרון 2:

- נניח שידוע לנו היכן נעשית פעולת הכפל האחרונה במימוש היעיל ביותר.

$$A_1 \cdots A_n = \left(A_1 \cdots A_t\right)\left(A_{t+1} \cdots A_n\right)$$

ואז אם נפתור את שתי תתי הבעיות הנ"ל אז יהיה לנו פתרון לבעיה כולה.

- הרעיון: אנחנו נפתור לא רק את הבעיה המקורית שלנו, אל לכל זוג אינדקסים $1 \leq l \leq k \leq n$ נמצא את הדרך היעילה ביותר מבחינה חישובית לחישוב המכפלה.
- רצינו למצו את דרך החישוב היעילה ביותר למציאת $A_1 \cdots A_n$ ובמקום זה אנו פותרים $\binom{n+1}{2}$ בעיות מהצורה $A_l A_{l+1} \ldots A_k$ (כלומר תת רצף של מטריצות).
- נגדיר c_{lk} כמחיר הזול ביותר לחישוב המכפלה $A_l \cdots A_k$, כאשר המטרה שלנו היא למצוא את c_{1n}.
באופן כללי ניתן לכתוב כי

$$c_{lk} = \min\left\{ c_{ls} + c_{sk} + \overbrace{\alpha_{l-1}\alpha_s\alpha_k}^{\text{Cost for multiplication}} \quad | l \leq s \leq k \right\}$$

- ולכן עם נתחיל עם c_{1n} (הגודל אותו אנו מחפשים) נוכל לעבור על כל האפשרויות של s, ואם c_{lk} מסויים לא ידוע אז נבצע את החישוב בו בצורה רקורסיבית ובחזרה מן החישוב נשמור תוצאה זו במערך דו מימדי.
  - ברור כי c_{kk} (איברי האלכסון) הם כמובן אפס, כי מדובר למעשה ברצף של מטריצה בודדת ולכן לא מכפילים אותה.
  - בנוסף אנו יודע את המחיר של כפל שתי מטריצות עוקבות בודדות והוא: $c_{l,l+1} = \alpha_{l-1}\alpha_l\alpha_{l+1}$.
- המחיר החישובי הוא $O\left(n^3\right)$ - מספר המספרים המחושבים הוא בערך n^2 וכל חישוב כזה אנו ממזערים על פני רשימה שאורכה לכל היותר n ולכן $O\left(n^3\right)$.
- דבר נוסף המתבקש שנברר הוא לא רק המחיר המיטבי, אלא גם המיסגור המיטבי. אך בכל שלב בו מצאנו את השילוב המינימלי, מצאנו את ה־s של החלוקה, ולכן בכל שלב אנו מחלקים את הרצף לשני חלקים, וחלוקה זו היא המיסגור.
בכל משבצת של המערך שבנינו נרשום גם את ערך האינדקס s המיטבי, ומתוך זה קל לגזור את המסגור האופטימלי.

3.2 בעיה מבילוגיה חישובית

DNA שהוא רצף המורכב מא"ב של 4 אותיות A, C, G, T שממנו ניתן להרכיב RNA המורכב מא"ב של A, C, G, U וממנו ניתן לבנות חלבון וכן הלאה. האותיות של ה־DNA הללו מתארגנות בזוגות $(C-G$, $A-T$ וכן הלאה), וכך נוצרת הסליל הכפול (Double Helix). תהליך דומה קורה גם עבור ה־RNA, ומתצפיות עולה כי ל־RNA יש מבנה "שניוני".

בעיה: אם ידועה לנו סדרת הבסיסים של מולקולת RNA מסויימת, נרצה לחשב את המבנה השניוני שלה.

העקרון היסודי: אנו מניחים שהמבנה השניוני של מולקולה נכונה, מוכדר ע"י כך שהוא המבנה בעל האנרגיה הנמוכה ביותר.
כאשר זוג בסיסים יוצר זיווג (כלומר $A-U$ ו־$C-G$) אנו מקטינים את האנרגיה של המולקולה.

המטרה: למצוא את זיווג הבסיסים המותר שבו מזווגים מספר מירבי של זוגות.

אילוצים: מהו זיווג מותר?

- כל בסיס עשוי להשתתף בזוג אחד לכל היותר.
- הזיווגים המותרים היחידים הם $A-U$ ו־$C-G$.
- ע"מ ששני בסיסים יזווגו, מרחקים ברצף צריך להיות לפחות 4.
- אין הצלבות (תמונה בטלפון)

פתרון: נניח שאורכה של מולקולת ה־RNA שלפנינו הוא nn רוצוננו למצוא את האנרגיה מינימלית ולכן את המבנה השניוני שלי.
בדומה למה שעשינו בבעיית כפל המטריצות, נגדיר $OPT\left(k,l\right)$ יהיה המספר המירבי של זוגות מזווגים (מותרים) במולקולת ה־RNA החלקית המתחילה במקום ה־k ומסתיימת במקום ה־l.
לכן, מטרתנו היא לחשב את $OPT\left(1,n\right)$, ונמצא זאת ע"י כך שנחשב את $O\left(n^2\right)$ המספרים ש־$OPT\left(k,l\right)$ כך ש־$1\leq k\leq l\leq n$, ולכן זמן הריצה הכולל הוא $O\left(n^3\right)$.

טענה 3.1

$$OPT\left(k,l\right)=\max\left\{ \overbrace{OPT\left(k,l-1\right)}^{*},1+\overbrace{\max_{t}\left[OPT\left(k,t-1\right)+OPT\left(t+1,l-1\right)\right]}^{**}\right\}$$

הוכחה: האפשרות ה־$(*)$ מייצגת את המקרה בו ויתרנו על הבסיס ה־l.
האפשרות ה־$(**)$ מייצגת את המקרה בו קיים חיבור בין המקום ה־t עד ה־l ואנו מחפשים את השילוב המקסימלי בין אפשרויות אלו, ובהנחה ומצאנו שילוב זה נוסף 1 עבור השילוב החדש שהרגע עשינו (בין t ל־l). ■

3.3 בעיית כדורי הבדולח

חידה: יש בניין בן n קומות, ויש k כדורי בדולח. מטרתינו לערוך ניסוי שיקבע מהי הקומה הגבוהה ביותר שממנה ניתן להשליך כדור בלי שיתנפץ.
יש איזשהו $D\in\left[1,n\right]$ לא ידוע כך ש־אם משליכים כדור מקומות $1,\ldots,D$ אז הוא נשאר שלם, ואם מעליו אז הוא ישבר.
מטרתינו למצוא מהו F תוך ביצוע מספר מזערי של ניסויים (ניסוי הוא השלכת כדור פעם אחת).

פתרון: (לא פתרון אופטמלי) נסמן ב־$\phi\left(k,n\right)$ את מספר הניסויים המזערי שמבטיח שנוכל לומר מהוא D.
עבור $k=1$ n $\phi\left(1,n\right)=$. ע"מ לקבוע את D ב־n ניסויים נעלה בקומות מעל קומה 1 וזה מרה שיש $\phi\left(1,n\right)\leq ni$.
$\phi\left(1,n\right)\geq n$ מראה שאם בתהליך הניסויי שלנו אנו פוסחים על קומה מסויימת, לא נפתור את הבעיה במקרה שפסחנו בדיוק על קומה D.
לא קשה לראות שאם $k=2$ ואנו מוכנים לבצע t ניסויים, נצליח לפתור את הבעיה בבנייה בגובה $n=\Omega\left(t^2\right)$, כלומר $\phi\left(2,n\right)=O\left(\sqrt{n}\right)$.
אם נבצע ניסויים בכפולות של $\sqrt{n}$ אז ברגע שבו הכדור ישבר נוכל לבצע חיפוש רגיל שלב שלב, ולכן

$$\phi\left(2,n\right)\leq2\sqrt{n}$$

כי באסטרטגיה שתיארנו, לכל היות נבע $\sqrt{n}$ ניסויים בכדור הראשון ו־$\sqrt{n}$ ניסיים בכדור השני ב־$\geq 2\sqrt{n}$.
ההתנהגות של ϕ ב־k קבוע כאשר $n \to \infty$ היא $\Theta\left(n^{\frac{1}{k}}\right)$.

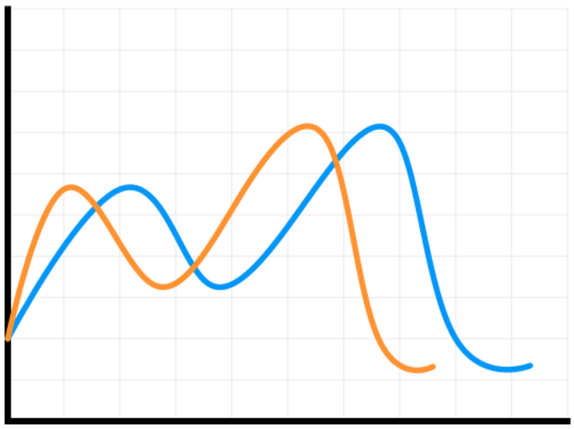
פתרון ממש: איך נחשב את $\phi\left(k,n\right)$?

$$\phi\left(k,n\right)=1+\min_{1\leq x\leq n}\left\{\max\left(\phi\left(k-1,x-1\right),\phi\left(k,n-x\right)\right)\right\}$$

X הוא הקומה שממנה מופל הכדור בפעם הראשונה בפתרון האופטמלי של הבעיה, אם הכדור התנפץ, אז אנו יודעים כי D נמצא מתחת ל־x ולכן נצטרך לפתור בעיה עם $k-1$ כדורים ו־$x-1$ קומות. אם הכדור לא נשבר, אז D מעל x, ועדיין יש לנו k כדורים ואנו צריכים לפתור את הבעיה של $n-x$ קומות.
כיוון שאנו לא יודעים את תוצאת הניסוי, עלינו לקחת את המספר המקסימלי מבין שתי הפתרונות בשביל שנוכל לפתור את הבעיה השלמה.
עתה, נצטרך לבדוק עבור כל קומה מה הפתרון המינימלי מבין כל הקומות (כלומר מה הצעד שיביא למינימום ניסויים).

3.4 עיבוד דיבור במחשב – אלגוריתם התאמת זמנים דינמית – DTW=Dynamic Time Warping

נניח וקיימת מערכת היודעת לזהות קול של דיבור, והמערכת אומנה לעבוד עבור הקלטה באורך מסויים, כיצד נוכל לזהות בין הקלטה זו לבין אותה הקלטה רק הנמשכת זמן אחר (מתיחה או כיווץ של הזמן).



איור 3.1: שני אותות זהים בזמנים שונים

איך מטפלים בבעיה? התבנית (מה שנשמר כחתימה בזיכרון) מתוארת ע"י סדרת מספרים ממשים $x_1,\ldots,x_N$ ומסנה להתאים לה סדרה $y_1,\ldots,y_M$.

דיסקרטיזציה: דוגמים את האות ברווחי זמן קצרים וכך יש לנו דרך להשוות את הסדרה $x_1,\ldots,x_n$ על מספר ממשיים עם הסדרה $y_1,\ldots,y_m$.
כאשר אם $m=n$ ו־$\forall i, x_i=y_i$ אז הסדרות זהות.

נניח שהחלטנו ש־$y_i \leftrightarrow x_i$ (בהנחה שאכן הסדרות מתאימות לפי הזמן), ואז קל לכמת את מידת השוני וההבדל הוא $|x_i-y_i|$.

אנו למעשה רוצים להתאים את הנקודות, כך שמותר ששתי נקודות יתאימו לאותה נקודה כל עוד אין הצלבות בין ההתאמות לפי השרטוט הבא:(תמונה במצלמה)

התאמה בין סדרת ה־x־ים לסדרת ה־y־ים איננה אלא מסילה מהנקודות $(1,1)$ שבמערכת הזה לנקודה (n,m) , ז"א יהיה עלינו להגדר מחיר לכל מסילה כזו ואז מטרתינו תהיה למצוא מסילה "זולה" ביותר כלומר "קצרה" ביותר.

העקרונות בהגדרת הבעיה הם:

1. על ההתאמה בין x_i, y_i נשלם מחיר $|x_i - y_j|$.
2. נרשה אך ורק צעדים $\rightarrow$ (פוסחים על x כלשהו), $\uparrow$ (פוסחים על y כלשהו) או $\nearrow$ (מתקדמים יחד ב־x וב־y).
בהתאם לזה נקבע מכיר c להליכה $\rightarrow$ או $\uparrow$ ונגבה מחיר אפס עבור הליכה $\nearrow$.
3. לכל צעד אסור ($\downarrow, \swarrow, \leftarrow, \nwarrow, \searrow$) נגדיר מחיר אינסופי.

פתרון הבעיה: במקום למצוא רק את מחיר המסילה (=מחיר ההתאמה המלאה) מ־$(1,1)$ ל־(m,n) אנו נמצא את כל mn המספרים $cost\,(i,j)$ כאשר $1 \leq i \leq m$ ו־$1 \leq j \leq n$ שזהו המחיר המזערים של התאמת הסדרה $x_1, \ldots, x_i$ לסדרה $y_1, \ldots, y_j$.

$$cost\,(i,j) = |x_i - y_j| + \min \{cost\,(i, j-1) + c, cost\,(i-1, j) + c, cost\,(i-1, j-1)\}$$

נסרוק את המערך הדו מימדי באופן אלכסוני (כאשר למעשה גם לפי שורות או עמודות היה ניתן לפתור ביעילות) ובזמן $O\,(mn)$ נפתור את הבעיה.

4 זרימה ברשתות – Flows in networks

נניח שיש רשת כבישים המתוארת ע"י גרף כיווני (לאו דווקא חסר מעגלים) עם נקודת מקור S ונקודת סיום, כאשר לכל צלע יש משקל המתאר את הקיבול של קטע הכביש (כלומר מספר הרכבים היכולים להימצא בו ברגע נתון).

הגדרה 4.1 רשת זרימה הוא גרף מכוון $G = (V, E)$ עם שני קודקודים מיוחדים $s \neq t \in V$ כאשר s נקרא מקור (source) ו־t נקרא הבור (sink).
כמובן מותאם לכל צלע $e \in E$ הקיבול שלה (capacity) $c\,(e) \geq 0$.

הערה 4.2 בשלב זה נניח כי לא נכנסות צלעות למקור $d^-\,(s) = 0$ וללא יוצאות צלעות מהבור $d^+\,(t) = 0$.

הגדרה 4.3 זרימה היא פונקציה $f : E \rightarrow \mathbb{R}^+$ המקיימת שתי דרישות:

1. אסור לעבור את הקיבול $\forall e \in E, c\,(e) \geq f\,(e) \geq 0$.
2. חוק קירכהוף (שימור החומר) – $\forall v \neq s, t, \sum_{\text{entering v}} f\,(e) = \sum_{\text{exiting v}} f\,(e)$.

הגדרה 4.4 השטף $\phi\,(f)$ של זרימה f מוגדר כ־$\phi\,(f) = \sum_{\text{exiting s}} f\,(e)$.

הבעיה: בהינתן הרשת נרצה למצוא זרימה המקבל שטף מירבי (כלומר זו בעיית אופטמיזציה).

תזכורת: כזכור בעיית אופטמיזציה מוגדרת ע"י קבוצה D ופונקציה $g : D \rightarrow \mathbb{R}$ והמטרה היא למצוא $\max_{x \in D} g\,(x)$.

הגדרה 4.5 בעיות הכרעה (Decision Problems) היא בעיה של כן או לא (למשל האם קיים פתרון אופטימלי? אך לא דרוש למצוא את הפתרון).

דוגמא: נתונה רשת G ונתון מספר K האם יש זרימה ב־G בעלת שטף $\leq K$?

הערה 4.6 בהינתן פתרון לבעיית הכרעה, ניתן להשתמש בה למציאת פיתרון אופטימלי (למשל ע"י חיפוש בינארי בתוך מכלול הפתרונות ע"י פסילת פתרונות לא טובים)

שאלה: איך ניתן לקבל עדות (witness) לכך שזרימה נתונה היא אופטימלית? (בהמשך ניראה כי מציאת עדות היא חלק מהגדרת בעיות NP).

הגדרה 4.7 אם $G=(V,E,s,t,c)$ היא רשת זרימה, אז חתך (cut) ב־G היא חלוקה $V=A\bigcup B$ (איחוד זר) כך ש־$s\in A$ ו־$t\in B$, והקיבול של חתך זה הוא

$$c(A,B)=\sum_{e:A\to B}c(e)$$

טענה 4.8 אם $G=(V,E,s,t,c)$ היא רשת זרימה, f זרימה ברשת זו, ו־(A,B) חתך ברשת, אז $c(A,B)\geq\phi(f)$ - כלומר כל חתך מספר חסם עליון לשטף של כל זרימה.

משפט 4.9 (משפט השטף והחתך – max flow min-cut theorem) בכל רשת זרימה יש זרימה מותרת f וחתך (A,B) כך ש־$\phi(f)=c(A,B)$.
כל זוג כזה הוא בהכרח אופימלי, ז"א ל־f שטף מירבי ול־(A,B) קיבול מזערי.

$$\max_f\phi(f)=\min_{V=A\bigcup B,A\bigcap B=\emptyset}c(A,B)$$

הערה 4.10 הבעייה של מציאת חתכים בעלי קיבולות קטן היא חשובה בפני עצמה.

רעיון כללי: (לא מדוייק - נתקנו בהמשך) איך נמצא זרימה בעלת שטף גדול? נצא מהזרימה $f\equiv0$.
בכל שלב נחפש מסילה מ־s ל־t ונזרים עליה איזשהו ערך שנוכל.
בכל מסילה שכזו יש צלע קריטית (בצוור בקבוק) וקיבולה הוא שיכתיב את הערך המירבי שניתן להזרים במסילה הזו.

תוכנית העבודה: בהינתן G וזרימה בה f, נרצה להגדיר רשת שיורית (residual) G_f כשנמצא מסילה מ־s ל־t ב־G_f נוכל להשתמש בה ע"מ לשפר את f.
אם e היא צלע ב־G ואם $c(e)>f(e)$, אז הגיוני לומר ש־e היא צלע ברשת השיורית G_f וקיבולה שם הוא $c(e)-f(e)>0$.
אם e היא צלע ו־$f(e)=\alpha>0$ נרצה להכליל ברשת השיורית גם את הצלע המכוונת $\overleftarrow{e}$ שהיא e אבל כיוון ההפוך וקיבולה יהיה α (כלומר ניתן להתייחס לכך שזורמים ברשת זרמים חיוביים וזרמים שליליים וסכומם הכולל הוא הזרם ממש בתוך הרשת).

דוגמא: נתונה רשת הזרימה הבאה:(סרטון)

הגדרה 4.11 אם P מסילה מכוונת מ־s ל־t בגרף עם קיבולים c לצלעות, אז נגדיר את הרוחב של P בתור $\min c(e)$.
לצלע המשיגה את $\min c(e)$ נקרא הצלע הקריטית ב־P (צוור הבקבוק של P).

הגדרה 4.12 תהיה G רשת ו־f זרימה בה, אז הרשת השיורית G_f היא רשת זרימה כך ש־$V(G_f)=V(g)$ כך שהמקור והבור ב־G_f הם s,t ו־$E(G_f)=\{\vec{e}\in E|c(e)>f(e)\}\bigcup\{\overleftarrow{e}\in E|f(e)>0\}$.
הקיבולים לצלע $\vec{e}$ הוא $c(e)-f(e)$ ב־G_f ועבור $\overleftarrow{e}$ יש קיבולות $f(e)$ ב־G_f.

הסכימה האלגוריתמית של Ford-Fulkerson: ע"מ למצוא זרימה אופטימלית ברשת נתונה:

1. צא מהזרימה $f \equiv 0$.
2. בכל איטרציה מצאה ב-G_f מסילה מ-s ל-t והזרם עליה ברוחבה, ז"א ככל שהצלע הקריטית מאפשרת.
3. עצור כאשר ב-G_f אין עוד מסילה מכוות מ-s ל-t.

בעיות בסכימה הזו:

1. יש דוגמאות (עם c שמקבלת ערכים אי רציונלים) שבהם הסכימה של FF אינה נעצרת כלל.
2. כאשר c מקבל רק ערכים שלמים (או רציונלים) אכן FF מסיימת את ריצתה, אבל זמן הריצה שלה אינו פולינומי באורך הקלט, כאשר הוא לכל היותר $O\left(\sum c\left(e\right)\right)$.
   רעיון הוכחה: מדי איטרציה השטף גדל לפחות ב-1 ובסה"כ השטף ברשת נתונה $\leq \sum c\left(e\right)$.
3. אין כאן הוראה ברור איזו מסילה P להעדיף.

שני מימושים קונקרטיים יעילים, בעלי זמן ריצה שהוא פולינומי באורך הקלט של סכימת FF שעליה נדבר:

1. נבחר מסילה בעלת רוחב מירבי (בערך).
2. (זהו אלגוריתם פולינומי חזק) אלגוריתם Edmords Karp - מצא את המסילה הקצרה ביותר מ-s ל-t.

4.13 תהיה $X \subset V$ ו-f זרימה אז נסמן ב-$f^{out}\left(X\right)=\sum_{e\in X,a\notin X,(e,a)\in E}f\left(\left(e,a\right)\right)$ ובאותו אופן $f\left(\left(a,e\right)\right)$ $\sum_{e\in X,a\notin X,(a,x)\in E}$ $f^{in}\left(X\right)$.

טענה 4.14 תהיה $A\subset V$ ו-$s\in A$ ו-$t\notin A$, ו-f זרימה אזי

$$\phi\left(f\right)=f^{out}\left(A\right)-f^{in}\left(A\right)$$

הוכחה: נשים לב ש-$\phi\left(f\right)=f^{out}\left(\{s\}\right)-f^{in}\left(\{s\}\right)$ וכמו כן, שימור החומר אומר שלכל $u\neq s,t$ מתקיים

$$f^{out}\left(\{u\}\right)=f^{in}\left(\{u\}\right)$$

נסכום את השוויון האחרון

$$\phi\left(f\right)=\sum_{v\in A}\left(f^{out}\left(\{v\}\right)-f^{in}\left(\{v\}\right)\right)=f^{out}\left(A\right)-f^{in}\left(A\right)$$

השוויון השמאלי מתקיים כי כמעט כל המחובר בסכום הם אפס, מלבד $v=s$ ולכן הסכום $\phi\left(f\right)=f^{out}\left(\{s\}\right)-f^{in}\left(\{s\}\right)$.
בסופו של דבר כל צלע היוצאת מ-A והנכנסת ל-A. ■

תרגיל: תהיה f הזרימה המקסימלית האפשרית במסילה מסויימת לפי הצלע הקריטית במסילה כלשהי, ונגדיר ממנה G_f, הרשת השיורית.
אם נחזור על התהליך ונמצא מסלול אחר מהמקור לבור, ונמצא זרימה g, אז אם נגדיר h כך שבצלעות שלא במסלול הוא מוגדר לפי f ובמסלול לפי $f+g$, אזי גם h זרימה מותרת ב-G והשטף שלה $\phi\left(f\right)+\gamma$ (כאשר γ הוא הקיבול של הצלע הקריטית במסלול ב-G_f).

תזכורת: לכל f זרימה ב-G ולכל $A\subset V$ נגדיר $f^{out}\left(A\right)$ בתור $\sum_{e\in A,b\notin A}f\left(e,b\right)$ ובאותו אופן עבור $f^{in}\left(A\right)$.

טענה 4.15 תהיה G רשת ו־(A,B) חתך, ו־f זרימה, אזי $\phi\left(f\right)=f^{out}\left(A\right)-f^{in}\left(A\right)$.

הערה 4.16 אם $A=\left\{ s\right\}$ אז זו ההגדרה $\phi\left(f\right)=\sum_{exit\,source}f\left(e\right)=f^{out}\left(\left\{ s\right\} \right)-\overbrace{f^{in}\left(\left\{ s\right\} \right)}^{=0}$.
נעיר עוד שאם $v\in V$ כך ש־$v\neq s,t$ אז $f^{out}\left(v\right)-f^{in}\left(v\right)=0$ (מחוק קירכהוף)

$$\sum_{v\in A}\left(f^{out}\left(v\right)-f^{in}\left(v\right)\right)=f^{out}\left(s\right)-f^{in}\left(s\right)+\overbrace{\sum_{v\neq s}\left(f^{out}\left(v\right)-f^{in}\left(v\right)\right)}^{=0}=\phi\left(f\right)$$

טענה 4.17 לכל רשת G ולכל זרימה f בה ולכל חתך (A,B) ב־G מתקיים ש־$\phi\left(f\right)\leq c\left(A,B\right)$.

הוכחה: מתקיים כי

$$\begin{aligned}\phi\left(f\right) & = & f^{out}\left(A\right)-f^{in}\left(A\right)\\ & = & \sum_{e:A\to B}f\left(e\right)-\sum_{e:B\to A}\overbrace{f\left(e\right)}^{\geq0}\\ & \leq & \sum_{e:A\to B}f\left(e\right)\\ & \leq & \sum_{e:A\to B}c\left(e\right)=c\left(A,B\right)\end{aligned}$$

■

הערה 4.18 נשים לב ש־$\phi\left(f\right)=c\left(A,B\right)$ אם"ם לכל $e:A\to B$ מתקים $f\left(e\right)=c\left(e\right)$ ולכל $e:B\to A$ מתקיים $f=0$.

משפט 4.19 תהיה G רשת ו־f זרימה מותרת בה, ונניח שברשת השירות G_f אין מסילה מכוונת מ־s ל־t, אזי יש חתך (A,B)ב־G המקיימת $C\left(A,B\right)=\phi\left(f\right)$.

מסקנה 4.20 הערות ומסקנות מתוך המשפט:

1. מהמשפט נובע שאם וכאשר אלגוריתם מטיפוס FF עוצר, אז הזרימה שבידינו היא אופטימלית, כי יש בידינו זרימה f וחתך (A,B) כך ש־$\phi\left(f\right)=c\left(A,B\right)$ וראינו שבמקרה כזה,הן f זרימה בעלת שטף מירבי והן (A,B) חתך בעל קיבול מזערי.
2. אם הקיבולים $c\left(e\right)$ שלמים, אז כל אלגוריתם מטיפוס FF מוצא זרימה אופטימלית בזמן $O\left(m\sum c\left(e\right)\right)$, כאשר m הוא מספר הצלעות בגרף.
בכל איטרציה של FF השטף גדל לפחות ב־1, ובכל מקרה וודאי $\phi\left(f\right)\leq\sum c\left(e\right)$ וביצוע איטרציה אחת של FF אורכת זמן של $O\left(m\right)$.

הוכחה: (המשפט) נראה שהטענה תקפה אם בוחרים את $A\subset V$ כאוסף כל הקודקודים ב־G_f שאליהם ניתן להגיע במסילה מכוונת מ־s.
$s\in A$ (ע"מ שתהיה מסילה) ו־$t\notin A$ כי ע"פ ההנחה אין מסילה מכוות מ־s ל־t.
כפי שראינו השוויון $c\left(A,B\right)=\phi\left(f\right)$ שקול לתכונות: (1) לכל $e:A\to B$ מתקיים $f\left(e\right)=c\left(e\right)$ (2) ולכל $e:B\to A$ מתקיים $f\left(e\right)=0$.
נוכיח כי מתקיימות התכונות:

1. תהיה $e=(u,v)$ כאשר $u\in A, v\notin A$ ונניח ש־$f(e)<c(e)$.
במקרה זה ניתן להגיע ב־G_f מ־s ל־v בניגוד להנחה ש־$v\notin A$.
העובדה ש־$u\in A$ פירושה שיש ב־G_f מסילה $s\to u$ ואם $f(e)<c(e)$ אז ב־G_f קיימת גם הצלע (u,v) רק בקיבול $c(e)-f(e)$, ולכן ניתן להמשיך צעד נוסף ולהגיע מ־s ל־v.
2. באותו אופן, אם $u\in A$ ו־$v\in B$ ו־$(v,u)\in E$ ומתקיים כי $f(e)>0$ אז נראה כי לא ייתכן כי $v\notin A$.
הסיבה לכך ש־$u\in A$ היא שיש ב־G_f מסילה מ־s ל־u, אבל אם קיימת הצלע (v,u) אז הצלע שייכת ל־G_f וקיבולה $f(e)>0$ ושוב ניתן לבנות מסילה מ־s ל־v.

■

משפט 4.21 (MFHC - Max Flow Min Cut) לכל רשת זרימה $\max\phi(f)=\min c(A,B)$.

הוכחה: נוכיח לפי מקרים:

- אם כל הקיבולים הם שלמים אנחנו כבר יודעים מהמשפט: f היא הזרימה שיש בידינו כאשר אלגוריתם FF עוצר וכפי שראינו זה קורה בזמן סופי.
- אם הקיבולים הם רציונליים, אז ניתן להסיק את המשפט מהמקרה הקודם, ע"י למשל הכפלת כל הגדלים במכנה המשותף של כל ה־$c(e)$.
- במקרה הכללי נקרב את $c(e)$ ע"י מספרים רציונליים, ונראה ש־

$$\forall\epsilon>0\exists f,(A,B),\phi(f)\geq c(A,B)-\epsilon$$

ע"י $\epsilon\to 0$ יתקבל המשפט.

■

מסקנה 4.22 (תכונת השלמות) אם $G=(V,E,s,f,c)$ היא רשת שבה כל הקיבולים הם מספרים שלמים, אז יש זרימה אופטימלית שבה כל ערכי הזרימה הם מספרים שלמים.

הוכחה: נניח שכל הקיבולים $c(e)$ הם מספרים שלמים, ונעקוב אחרי התקדמותו של אלגוריתם FF (כלשהו).

1. מכיוון שערך צוואר הבקבוק בכל מיסלת הרחבה בכל צעד של האלגוריתם הוא מספר טבעי, אז כל ערכי f לאורך כל האלגוריתם הם מספרים שלמים.
2. כאשר האלגוריתם עוצר (כוכפי שראינו הוא עוצר בזמן סופי) ומתקבלת זרימה אופטימלית וכל ערכי שלמים.

■

4.0.1 כמה מילים על תכנון לינארי (Linear Programming)

כזכור, בעיית אופטימיזציה היא שבהינתן $g:D\to\mathbb{R}$ אז מחפשים $\max_{x\in D}g(x)$.

נצטמצם למקרה שבו $g:\mathbb{R}^n\to\mathbb{R}$ היא פונקציה לינארית, כלומר $x\mapsto\langle x,a\rangle$ (מכפלה פנימית) כאשר $a\in\mathbb{R}^n$ הוא ווקטור כלשהו.
גם הקבוצה $D\subset\mathbb{R}^n$ מוגדרת ע"י מערכת של משוואות ואי שוויונים לינארים ־ לבעייה זו קוראים LP.
קיימים אלגוריתמים יעילים לפתרון בעייה זו.

נשים לב כי בעיית זרימה היא למעשה בעיית LP. זרימה מותרת מוגדרת ע"י משוואות $\forall v, f^{out}(e)-f^{in}(e)=0$ ואי שוויונות לינאריים $0\leq f(e)\leq c(e)$.

משפט השטף והחתך הוא מקרה פרטי של משפט הדואליות בתכנון לניארי.

4.1 מימושים של אלגוריתם FF

4.1.1 אלגוריתם מדורג – Scaling

הרעיון: באלגוריתם המדורג אנו מחפשים מסילות הרחבה בעלות קיבול גבוה (ז"א ערך צוואר הבקבוק הוא גדול).

בשביל למצוא את המסילה הרצויה, נגדיר גודל Δ ובכל שלב נשאיר בגרף רק צלעות בעלות קיבול $\leq \Delta$ ונשאל האם יש עדין מסילה מ־s ל־t. אם כן, אז נעל את Δ, אם לא אז נוריד את Δ.
בפועל נבצע על הערך Δ חיפושה בינארי על חזקות של 2 ורק בהן נפעל.
כמובן, מדי צעד הרחבה אז נגדיל את השטף $\leq \Delta$ והאלגורית סיהיה יעיל מהסיבות הבאות:

1. כאשר הערכים ש־Δ מקבל הם חזקות של 2 המתחילות ב־$\max c(e)$ (מעוגל מטה לחזקת 2). ויורד מדי צעד לחצי ערכו $\Delta \leftarrow \Delta/2$, ולכן מספר הפאזות (פרקי הזמן באלגוריתם שבהם Δ הוא ערך מסויים $\leq \lceil \log_2 C \rceil$ כש־$C = \sum c(e)$).

הגדרה 4.23 $G_f(\Delta)$ זו הרשת המתקבלת מ־G_f ע"י השמטת כל הצלעות שקיבולת $< \Delta$.

סכמטי של האלגוריתם:

- אתחל את Δ ל־$\max(e)$ מעוגל מטה לחזקה הגבוה ביותר של 2 האפשרית.
- אתחל את $f \equiv 0$.
- ברשת $G_f(\Delta)$ מצא מסילת הרחבה (מסילה מ־s ל־t) ועדכן את f והמשל כך עוד יש מסילה מ־s ל־t ב־$G_f(\Delta)$.
- הקטן את $\Delta \leftarrow \Delta/2$ וחזור עד $\Delta = 1$.

טענה 4.24 האלגוריתם הנ"ל מוצא זרימה אופטימלית.

הוכחה: בגלל שהוא עוצר כאשר $\Delta = 1$ ובנוך מה שהוכחנו על אלגוריתם FF באופן כללי. ∎

טענה 4.25 האלגוריתם הזה רץ בזמן $O(m^2 \log C)$, כאשר m הוא מספר הקודקודים.

טענה 4.26 יש לאלגוריתם $O(\log C)$ פאזות.

הוכחה: מספר הערכים השונים שהפרמטר Δ עשוי לקבל הוא $\leq \lceil \log_2 C \rceil$, כי ערכו ההתחלתי $< 2\Delta$ ומדי צעד Δ מוכפל ב־$\frac{1}{2}$ והאלגוריתם נעזר כאשר $\Delta = 1$.
בנוסף, לאלגוריתם יש לכל היות $\log_2 C$ פאזות (תקופות שבהן Δ קבוע). ∎

טענה 4.27 כל פאזה אורכת זמן $O(m^2)$.

הוכחה: ניתן למצוא מסילת הרחבה בזמן $O(m)$, ולכן די להוכיח שעבור ערך של Δ מספר צעדי ההרחבה שנעשה הוא $O(m)$.
מאנחנו נראה שכאשר סיימנו פאזה נתונה Δ, אז השטף של הזרימה שבידינו רחוק מהאופטימום בלכל היותר $m\Delta$.
נראה זאת ע"י כך שנמצא חתך שקיבולו קרוב לשטף של הזרימה בידינו עד כדי ההפרש של $m\Delta$.
מדי צעד הרחבה בפאזה זו, אנו מגדילים את השטף לפחות ב־Δ ולכן מספר הצעדים הוא $O(m)$. ∎

טענה 4.28 בכל פאזה אנו מבצעים $O(m)$ צעדי הרחבה.

הוכחה: כפי שנראה, בסוף הפאזה ה־2Δ הזרימה שבידינו מרחקה מהאופטימום הוא $\geq m\Delta$.
שטף הזרימה כמובן אינו יכול לעלות על האופטימום והוא גדל מדי צעד הרחבה ב־$\leq \Delta$.
איך מראים שהשטף של הזרימה שבידינו f קרוב עד כדי $m\Delta$ לאופטימלי?
אנו מוצאים חתך (A,B) כך ש־$\phi(f) \geq c(A,B) - m\Delta$, ואכן

$$\begin{aligned}
\phi(f_\Delta) &= f_\Delta^{out}(A) - f_\Delta^{in}(A) \\
&= \sum_{e \text{ is out A}} f_\Delta(e) - \sum_{e \text{ is in A}} f_\Delta(e) \\
&\overset{*}{\geq} \sum_{e,out} (c(e) - \Delta) - \sum_{in} \Delta \\
&\geq c(A,B) - m\Delta
\end{aligned}$$

הוכחת ה־$(*)$: מתקיים כי:

1. לכל צלע e שיוצאת מ־A מתקיים כי $f_\Delta(e) \geq c(e) - \Delta$
2. לכל צלע שנכנסת ל־A מתקיים כי $\Delta \geq f_\Delta(e)$.

נתבונן בצלע $u \in A, v \notin A$ ו־$(u,v) \in E$ אז:

1. אילו היה $f_\Delta(e) < c(E) - \Delta$ הייתה ב־G_f צלע מכוונת מ־u ל־v בקיבול של $c(e) - f(e)$. היות שמספר זה $< \Delta$ היתה הצלע שנכנסת גם ב־$G_f(\Delta)$ ולכן גם $v \in A$ בניגוד להנחה.
2. כנ"ל עבור $e = (v,u)$. אילו היה $f_\Delta(e) > \Delta$ אז הצלע (v,u) הייתה ב־G_f ולכן הקיבול גדול מ־Δ ולכן הייתה שייכת גם ל־$G_f(\Delta)$ ולכן $v \in A$ בניגוד להנחה.

■

4.1.2 אלגוריתם Edmonds Karp

הרעיון: באלגוריתם EK, בכל שלב נשתמש במסילת ההרחבה הקצרה ביותר.

משפט 4.29 זמן הריצה של אלגורית EK הוא $O\left(nm^2\right)$.

הגדרה 4.30 אומרים שצעד הרחבה מסויים ממצה צלע נתונה e, אם אחד מהבאים מתקיים:

1. הוא מגדיל את הזרימה בה ל־$c(e)$.
2. הוא מוריד את הזרימה בה ל־0.

האבחנה הבסיסית: תהיא P המסילה הקצר ביותר $s \to t$ ב־G_f שאותה בחר EK ותהיה g הזרימה המעודכנת.
תהיה Q מסילה קצרה ביותר ב־G_g אילו המצב היה $|Q| \leq |P|$ (בניגוד לטענתינו) אז בהכרח יש קונפליקט בין P ו־Q, כלומר יש צלע e ששתי המסילות מבקרות בה אבל בכיוונים מנוגדים.

הוכחה: די להראות שהאלגוריתם מבצע $O(mn)$ צעדים של מסילות הרחבה (כי ניתן למצוא מסילת החרב קצרה ביותר בזמן $O(m)$).
טענה זה מסתמכת על שתי טענות עזר: (1) במהלך האלגוריתם המרחק מ־s ל־t ב־G_f אינו יורד, ו־(2) בין שני צעדים שבהם אותה הצלעה e מתמצה המרחק הנ"ל גדל ממש.

יש לכל היותר n פעמים שבהן מרחק s,t ברשת השיורית עולה, ונעיר שבין שתי עליות עוקבות כאלה, אנו מבצעים לכל היותר m צעדי הרחבה ־ מדוע? בכל צעד הרחבה אנו ממצים לפחות צלע אחת (הצלע הקריטית במסילת ההרחבה), אילו היו יותר מ־m צעדי הרחבה היינו מוצאים שני צעדים שבהם אותה צלע e מוצתה בלי שחלה עליה במרחק ובניגוד לטענה 2. ■

הרעיון להוכחת טענה 1: נניח ש־P הקצרה ביותר ב־G_f והזרימה החדשה אחרי השלב היא f'.
אם Q מסילה ב־$G_{f'}$ ו־$|P|>|Q|$, בהכרח Q אינה עומדת לרשותינו ב־G_f ולכן יש בה צלעות שאינן ב־G_f. צלע כזו חייבת להיות חלק מ־P, ועל צלע זו P,Q עוברים בכיוונים מנוגדים.
המצב שבו המרחק מ־s ל־t דווקא קטן אחרי איטרציה כלשהי של EK, מחייב שיהיו שתי מסילות EK עוקבות הנמצאות בקונפליקט ־ כלומר P,Q עוברות באותה הצלע אך בכיוונים מנוגדים.

טענה 4.31 יהיו $r' > r$ שני זמנים, ויהיו Q,P מסילות ההרחבה שבהן בוחר אלגוריתם EK בזמנים אלה, ונניח ש־P,Q נמצאות בקונפליקט.
נניח שלכל מסילת הרחבה R (של EK) הנבחרת בזמן בינים $r' > r'' > r$ המסילה R אינה בקונפליקט לא עם P ולא עם Q.
אזי $|Q| > |P|$.

הוכחה: נוכיח בשלבים: (הוכחה לא שלמה... אני אשלים בהמשך)

1. תחילה נראה עבור המקרה הפרטי הבאה:
ומתקיים כי $b_0 \geq a_0 + 1$ ו־$b_1 \geq a_1 + 1$ כאשר $|P| = a_0 + a_1 + 1, |Q| = b_0 + b_1 + 1$ ולכן

$$|Q| - 1 = b_0 + b_1 \geq a_0 + a_1 + 2 > |P|$$

מדוע האופציה לנוע דרך b_0 ניתנת לבחירה? אחרת יש צלעות שאינן קיימות ברשת השיורית בזמן r שבו אנו בוחרים ב־P.
ע"מ שצלע כזו תיווצר דרוש איזשהו קונפליקט בדרך עד Q וע"פ ההנחה אין R ביניים היוצרת קונפליקט כזה.

2. המקרה הכללי דומה. נסתכל על :

■

מסקנה 4.32 מתוך הטענה הנ"ל נובעת טענה 1 מההוכחה.

הוכחה: נביט על שתי איטרציות עוקבות של EK ($r' = r + 1$).
אם P,Q אינן בקונפליקט אז $|Q| \geq |P|$ כי גם Q עומדת לרשותינו בזמן r ואף על פי העדפנו את P.
אם P,Q בקונפליקט אז הנחות הטענה מתקיים באופן ריק (אין זמן ביניים r'') ולכן $|Q| > |P|$ כנדרש. ■

מסקנה 4.33 מתוך הטענה הנ"ל נובעת טענה 2 מההוכחה.

הוכחה: בין שני מיצויים עוקבים של e, חייב להיות איזשהו זוג מסילות בקונפליקט P,Q.
נתבהונן בזוג כזה שפרק הזמן בינהם באלגורית הוא קצר ביותר. בכלל המינימליות, אין R בינהם המצוי בקונפליקט עם P או עם Q, ולכן מהטענה מתקיים כי $|Q| > |P|$. ■

5 קירובים לבעיות NP קשות

הגדרה 5.1 המחלקה של הבעיה ה־NP קשות כוללת אלפי בעיות הכרעה שלגביהן ידועים שני דברים:

1. לא ידועים אלגוריתמים פולינומים לאף אחת מהן.
2. אילו היה ידוע על אלגוריתם פולינומי לאחת מהן, אז היה קיים אלגוריתם לכל הבעיות במחלקה זו.

אחד העיסוקים המרכזיים בתחם האלגוריתמים הוא למצוא שיטות קירוב יעילות לבעיות האופטימיזציה המתאימות לבעיות NP־שלמות.

5.1 בעיית הסוכן הנוסע ־ Travelling Salesman Problem

הבעיה: נתון גרף עם משקולות חיוביים (או אינסוף ־ כלומר נתק) ואנו מחפשים את המסילה הקצרה ביותר כך שבכל קודקוד היא מבקרת פעם אחת ומשקלה הכולל מזערי.

בעיית הכרעה: (NP קשה) האם בתוך גרף יש מעגל המילטוני ־ מסילה סגרה המבקרת בכל קודקוד בדיוק פעם אחת.

ידוע: אם $P \neq NP$ (P היא המחלקה שאת הבעיות ניתן לפתור ביעילות פולינומית) אז אין ל־TSP קירוב c שניתן למצוא בזמן פולינומי לכל $c > 0$ ממשי ־ כלומר בהינתן $c > 0$ אין אלגוריתם פולינומי המקבל את הקלט הנ"ל ומוצא בו מסילה סגורה במחיר לכל שהוא לכל היותר $c \cdot OPT$.

בעיה מצומצמת: אם המשקולות $w : E \to \mathbb{R}^{+}$ מקיימים את אי שוויון המשולש, אז קוראים לבעיית הסוכן הנוסע ΔTSP, כלומר

$$\omega\left(x,y\right)+\omega\left(y,z\right)\geq\omega\left(x,z\right)$$

נראה: אלגוריתם ל־2 קירובי של בעיית ה־ΔTSP ־ כלומר יש אלגוריתם המוצא בזמן פולינומי בהינתן גרף משוקלל שבו ω מקיימת את אי שוויון המשולש מסילה סגורה העוברת פעם אחת דרך כל הקודקודים ומחירה לכל היותר $2 \cdot OPT$.

טענה 5.2 לכל קלט לבעיית ה־TSP, $OPT > MST$.

הוכחה: OPT מחברת בין כל הקודקודים, ולכן אם נחסיר ממנו צלע אז נקבל עץ פורש (אולי מינימלי) ולכן כשנחזיר את הצלע נקבל את אי השוויון. ■

האלגוריתם: נבנה MST (יחסית למשקולת w).
נערוך הליכה בגרף היוצאת מהשורש וחוזרים אליו, וסורקת כל צלע של העץ הפורש המינימלי פעמיים.

היתרונות:

1. ביקרנו בכל קודקוד של הגרף.
2. מחיר הטיול הוא $2 \cdot MST$ ולכן הוא קטן מ־$2 \cdot OPT$.

החסרון: מבקרים באותו הקודקוד יותר מפעם אחת.

לאור החסרון נרצה לתקן את המסלול בעזרת אי שוויון המשולש כך ש:

1. המחיר לא יגדל.
2. בסופו של דבר יהיה בידינו מסלול סוכן נוסע העובר בכל קודקוד בדיוק פעם אחת.

נניח כי קיים מסלול $x \to y \to z$ כך שב־y מבקרים יותר מפעם אחר, אז נחליף את המעבר ב־$x \to z$ ומא"ש המשולש נקבל כי $\omega(x,z) \le \omega(x,y) + \omega(y,z)$ ולכן רק הקטנו את אורך המסלול, ועדיין אנו מבקרים בכל הנקודות.
כל עוד יש קודקודים שבהם ביקרנו יותר מפעם אחת, נפסח עליהם בדיוק באותו אפון בלי להגדיר את מחיר המסלול.

הערה 5.3 ניתן למצוא באופן יעיל קירוב של $3/2$ לבעיית ה־ΔTSP.
נניח כי מצאנו MST שהוא מסילה, אז במקרה זה אפשר פשוט להוסיף את הצלע בין תחילת המסילה (L) לסופה(F), ומאי שוויון המשולש אנו יודעים כי גודל הצלע הזו היא לכל היותר כמשקל כל ה־MST.
נשים לב כי ניתן להראות כמשקל הצלע הנוספת היא לכל היותר חצי מהמשקל של המסלול השלם של האלגוריתם האופטימלי.
מכיוון שהאלגוריתם המינימלי עובר בין F ל־L, אבל המרחק במסלול האופטימלי הוא לכל היותר $\frac{1}{2}OPT$, ולכן מאי שוויון המשולש קיבלנו את החסם העליון.
בסה"כ קיבלנו כי המעגל שמצאנו הוא פי $3/2$ מהפתרון האופטימלי.
ממקרה זה נוכל לבנות את האלגוריתם הבא:

1. בונים MST ע"פ משקולות העץ.
2. בונים זיווג M במחיר מזערי בין הקודקודים ב־T שדרגתם אי זוגית (לא למדנו בהרצאות, אבל קיים אלגוריתם יעיל לפתרון הבעיה).
3. הגרף $T+M$ הוא גרף אוילריני, כלומר לכל קודקוד יש דרגה זוגית, ויש בו מעגל אוילר C המבקר בכל צלע של $M+T$ בדיוק פעם אחת.
4. מבצעים על C קיצורים באמצעות שימוש באי שוויון המשולש, ומקבלים מעגל סוכן נוסע במחיר $\le \omega(T) + \Omega(M)$.
5. אבל $\omega(T) = MST < TSP$ ולכן $\omega(M) \le \frac{1}{2}OPT$ ע"י שיקול דומה כמו במקרה שהראינו לעיל.

5.2 בעיית VC - Vertex Cover

בתרגולים דיברנו על בעיית ה־VC שהיא NP קשה.

הקלט: גרף $G=(V,E)$.

הפלט: קבוצה קטנה ביותר של קודקודים $S \subseteq V$ כך שלכל צלע $E \ni e = (x,y)$ מתקיים כי $\{x,y\} \bigcap S \neq \emptyset$.

5.2.1 בעיית הכיסוי הקבוצתי (Set Cover Problem)

הקלט: אוסף תת קבוצות $A_1,\dots,A_m$ של קבוצת X, כך ש־$|X|=n$, וכך ש־$X = \bigcup_{i=1}^m A_i$.

הפלט: תת קבוצה קטנה ביותר של אינדקסים $J \subset \{1,\dots,m\}$ כך ש־$X = \bigcup_{j\in J} A_j$.

הערה 5.4 בעיה זו היא NP קשה.

אלגוריתם חמדן: צא מ־$J=\emptyset$ ובכל צעד הוסף ל־J את A_i שמגדיל ככל האפשר את $\left|\bigcup_{j\in J} A_j\right|$.

משפט 5.5 האלגוריתם החמדן הנ"ל לבעיית הכיסוי הקבוצת נותן קירוב של $O(\log n)$ לאופטימלי, כלומר

$$|J_{Greedy}| \leq (\ln n + 1)\,|J_{OPT}|$$

הערה 5.6 ידוע שבהנחות סבירות ($P \neq NP$) אין קירובים טובים יותר שניתן למצוא בזמן פולינומי.

הוכחה: (המשפט) נסמן ב־$C(n,k)$ את המחיר המירבי שעלול האלגוריתם החמדן לשלם בבעיית ה־SC כשגודל קבוצת הבסיס הוא n והפתרון האופטימלי משתתפות בדיוק k קבוצות, ולכן צריך להראות ש־

$$c(n,k) \leq (\ln n + 1)\,k$$

טענת עזר: $c(u,k) \leq 1 + c\left(u\left(1-\frac{1}{k}\right),k\right)$.
הוכחה: ע"פ ההנחה יש איזשהן k קבוצות באוסף שאיחודן הוא כל קבוצת הבסיס, ולכן בפרט לפחות אחת מ־k האלה הכוללת לפחות $\frac{1}{k}$ מאיברי קבוצת הבסיס, אבל האלגוריתם החמדן בוחר תמיד בקבוצה המחסה מספר איברים המירבי של האיברים החדשים, ולכן באחת כזו הוא יחבר וינסה לפחות $\frac{1}{k}$ מאיברי קבוצת הבסיס, ולכן מספר האיברים שאותם עלינו לכסות יורד מ־u למספר שהוא לכל היותר $u-\frac{u}{k}=u\left(1-\frac{1}{k}\right)$, כאשר כבר בחרנו קבוצה 1, ולכן

$$c(n,k) \leq 1 + c\left(u\left(1-\frac{1}{k}\right),k\right)$$

באמצעות הטענה הנ"ל נוכל למצוא כי

$$\begin{aligned} c(n,k) &\leq 1 + c\left(n\left(1-\frac{1}{k}\right),k\right) \leq 2 + c\left(n\left(1-\frac{1}{k}\right)^2,k\right) \\ &\leq \ldots \leq t + c\left(n\left(1-\frac{1}{k}\right)^t,k\right) \end{aligned}$$

ולכן נרצאה למצוא מהו ה־t המינימלי כך ש־$n\left(1-\frac{1}{k}\right)^t < 1$ ומתקיים כי

$$\begin{aligned} n\left(1-\frac{1}{k}\right)^t &\leq ne^{-\frac{t}{k}} < 1 \\ \Rightarrow \ln n &\leq \frac{t}{n} \end{aligned}$$

ולכן מספיק $t > k\log n$. ■

הערה 5.7 בעיית הכיסוי הקבוצתי שקולה (בשינוי פרמטרים) לבעיית אופטימיזציה חשבוה אחרת הקרוייה Hitting Set (קבוצה פוגעת).
נגדיר מטריצה

$$M_{i,j} = \begin{cases} 0 & i \notin A_j \\ 1 & i \in A_j \end{cases}$$

כאשר $A_1,\ldots,A_m$ הן קבוצות אינדקסים $X=\{1,\ldots,n\}$, ולכן M מגדירה עבור כל מטריצה אילו איברים של X שייכים לה.

עתה נרצה לבחור את הקבוצה המינימלת של A_i כך שהם מכסים את כל X.
הנחה: אין ב־M שורה של אפסים (כלומר $A_i \neq \emptyset$).
כמובן שחוץ משינוי באינטרפרטציה (מהי קבוצה, מהו איבר) אין הבדל אם נשאל את אותה השאלה בחילוף שורות ועמודות (זוהי בעיית ה־Hitting Set).
נתונה מטריצה של $0,1$ ללא עמודה של אפסים ־ מצא מספר מזערי של שורות הפוגעול בכל עמודה.
כלומר בעיית ה־Hitting Set בניסוח הקבוצתי היא: נתון אוסף של קבוצות לא ריקה $B_1,\ldots,B_l \subset Y$, מצא תת קבוצה קטנה ביותר של $H \subset Y$ כך שלכל $B_i \bigcap H \neq \emptyset$ ־ כלומר במקום לבחור שורות/עמודות נבחר קבוצת איברים שכל קבוצה פוגעת לפחות באחד מהם.

5.3 בעיית ה־Max Cut

הבעיה: נתון גרף $G=(V,E)$ חסר משקולות ומחפשים חלוקה של $V=A\bigcup B$ (איחוד זר) כך שמספר הצלעות $(x,y)\in E$ כך ש־$x\in A, y\in B$ הוא מירבי.

הערה 5.8 קיים אלגוריתם של Goeman, Williamson המוצא חתך שגודלו הוא לפחות $0.87\cdot OPT$.

הגירסה המשוקללת: נתונים משקולות $W_{ij}\geq 0$ כך ש־$W_{ij}=W_{ji}$ ולכל i מתקיים $W_{ii}=0$.
נדרש למצוא חלוקה $\{1,\ldots,n\}=A\bigcup B$ כך ש־$\sum_{i\in A,j\in B} W_{ij}$ מקסימלי.

הערה 5.9 נשים לב כי תנאי הכרחי לאופטימליות הוא שעבור שינויים קטנים, ערך הפונקציה רק יורד מהנקודה החשודה לאופטימלית.

נניח כי החלוקה אופטימלית ובנוסף מתקיים גם

1. $d_Y(x)\geq d_X(x)$ לכל $x\in X$ ($d_X(y)$ הוא מספר השכנים בקבוצה X של הקודקוד y)
2. $\forall y\in Y, d_X(y)\geq d_Y(y)$.

נסכם את 1 על פי x ו־2 על פי y וקיבלנו כי

$$e(X,Y)=\sum_{x\in X} d_Y(x)\geq \sum_{x\in X} d_X(x)=2e(x)$$

כלומר $e(X,Y)$ גדול או שווה לפעמים מספר הצלעות כך ששני הקודקודים ב־X.
באותו האופן נקבל כי $e(X,Y)\geq 2e(Y)$ ולכן

$$\begin{aligned} e(X,Y) &\geq (e(x)+e(y)) \\ 2e(X,Y) &\geq e(X,Y)+e(x)+e(Y)=|E| \end{aligned}$$

כלומר אם מתקיימות ההנחות הרשומות, אז בחתך X,Y נמצאות לפחות חצי מצלעות הגרף.
לכן אם ניתן (באופן יעיל) למצוא חלוקה $V=X\bigcup Y$ כך שמתקיימים התנאים הנ"ל, אז יש בידנו חתך המקרב את max_cut עד כדי גורם ≤ 2.

5.3.1 בניית חלוקה של V המקיימת את ההנחות

האלגוריתם: צא מחלוקה כלשהי אם יש קודקוד ב־X המפר את תנאי (1), העבר אותו ל־Y, ובאותו אופן אם יש קודקוד ב־Y המפר את תנאי (2) אז העבור אותו ל־X.

טענה 5.10 הלאגוריתם מגיע תוך $\leq O(m)$ צעדים לחלוקה המקיימת את התנאים (m הוא מספר הקודקודים).

הוכחה: מפני שבכל צעד שבו אנו מעבירים מצד לצד קודקוד המפר את (1) או את (2) אנו מגדילים את $e(X,Y)$ ב־1 לפחות, ולכן לא יהיה יותר מ־m צעדי העבר כאלו. ■

5.3.2 אלגוריתם הסתברותי המוצא חתך בגרף המכיל לפחות חצי מצלעות הגרף

האלגוריתם: מטילים מטבע לכל קודקוד $v \in V$.
אם יוצא ראש את מכניסים את v לקבוצה A, אחרת לקבוצה B.

החתך המתקבל הוא מקרי כמובן - מהי תוחל הגודל שלו?
$\Omega = \{H,T\}^V$ ובהסתברות של $\frac{1}{2^n}$ לכל מאורע אלמנטרי, ונגדיר $X : \Omega \to \mathbb{N}$ משתנה מקרי כך ש-$X(\omega) = e(A,B)$.
מהי התוכל של X? כלומר $E(X)$?

$$X = \sum_{\omega \in E} X_{uv}$$

כאשר $X_{uv}(\omega)$ שווה לאפס אם שני הקודקודים u ו-v באותה הקבוצה בחתך המושרה ע"י ω, ואחד אחרת.
בגלל הלינארית של התוחלת נקבל כי

$$\begin{aligned} E(X) &= \sum_{umv \in E} E(X_{uv}) = \sum_{u,v \in E} P(\text{u and v were sapparated}) \\ &= \frac{1}{2}|E| \end{aligned}$$

ולכן מצאנו אלגוריתם לינארי המוצא חתך כנ"ל.

5.3.3 אלגוריתם להעשרה

יש אלגוריתם (של גומנס ווילימסון) שמוצא באופן יעיל חתך שהוא $0.87 \cdot MAXCUT \leq$.
ניסוח אלטרנטיבי לבעיית החתך המקסימלי הוא

$$MAXCUT = \max_{x_1,\ldots x_n \in \{-1,1\}} \sum_{i,j} \frac{1 - x_i x_j}{2} W_{ij}$$

נוכל לכתוב בעייה זהו גם עבור $y_1,\ldots,y_n$ וקטורי יחידה ואז

$$\max_{x_1,\ldots,x_n \in \{-1,1\}} \sum_{ij} \frac{1 - x_i x_j}{2} W_{ij} \geq 0.87 \overbrace{\max_{y_1,\cdots y_n} \sum_{ij} \frac{1 - \langle y_i, y_j \rangle}{2} W_{ij}}^{\text{V-MaxCut}}$$

כאשר קל לראות ש- $VMAXCUT \geq MAXCUT$ כי אם נגביל את עצמנו ל-y-ים מהצורה $(\pm 1, 0, 0, \ldots)$ (כזכור y הוא וקטור) אז חזרנו לבעיית ה-$MAXCUT$ הרגילה.
ההוכחה של החסם התחתון ל-$MAXCUT$ לא תינתן בקורס זה.

5.4 בעיית Max 3SAT

רקע: נתונה נוסחה בצורת CNF, כלומר נתונה פונקציה $f(x_1,\ldots,x_n)$ המקבלת משתנים בוליאניים (אמת או שקר) שהיא מהצורה:

$$f(x_1,\ldots,x_n) = C_1 \wedge C_2 \wedge \ldots \wedge C_m$$

כך ש-C_i נקראת פסוקית (clause), והיא מהצורה $x_i^{\epsilon_1} \vee x_{i_2}^{\epsilon_2} \vee \ldots \vee x_{i_k}^{\epsilon_k}$ כך ש- $\epsilon_i \in \{1,-1\}$ (כלומר אפשר לשלול משתנה).

בעיה: בעיית הספיקות (SAT) - נתונה f כנ"ל בצורת CNF. האם יש ערכי אמרת למשתנים כך ש-$f = true$?
(ז"א האם יש ערכי אמת/שקר למשתנים כך שכל פסוקית C_i מקבל ערך אמת.

הערה 5.11 בעיה זו היא NP קשה.

מקרה פרטי: (וגם הוא NP קשה) הוא בעיית ה-$3SAT$ שבו בכל פסוקית יש בדיוק 3 משתנים.

בעיית אופטימיזציה: נתונה נוסחה ב-$3CNF$, וצריך למצוא ערכי אמת למשתנים כך שמספר מירבי של פסוקיות יקבלו את הערך אמת.

הערה 5.12 זו בעייה קשה ונראה אלגוריתם הסתברותי למציאת קירוב בשבילה.

האלגוריתם: הצבה מקרית של ערכי אמת.

טענה 5.13 שתי טענות

1. תוחלת מספר הפסוקיות שיסופקו בדרך זו היא בדיוק $\frac{7m}{8}$.
2. ההסתברות שנקבל לפחות $\frac{7m}{8}$ פסוקיות מסופקות היא $\le \frac{1}{m} \cdot \Omega$ (בעצם $\frac{8}{m}$).

תזכורת מהסתברות: אם מבצעים ניסויים ב"ת כשהסתברות ההצלחה בניסוי בודד היא p, אז תוחלת מספר הניסויים שלנו לבצע עד הצלחה היא $\frac{1}{p}$.

הוכחה: נוכיח את הטענות:

1. נגדיר $\Omega = \{T, F\}^n$ ומשתנה מקרי X המונה את מספר הפסוקיות המסופקות.
נפרק את המשתנה ע"י $X = \sum X_j$ כש-X_j הוא מ"מ מציין

$$X_j(\omega) = \begin{cases} 1 & \text{The j-th clause is true} \\ 0 & \text{Otherwise} \end{cases}$$

בכלל הלינארית של התוחלת

$$\begin{aligned} E(X) &= E\left(\sum X_j\right) = \sum EX_j \\ &= \sum_j \overbrace{P(X_j = 1)}^{\frac{7}{8}} \\ &= m\frac{7}{8} \end{aligned}$$

2. יהיה $p_0, \dots p_m \ge 0$ כש-$p_i \ge 1$ ההסתברות שסיפקנו בדיוק i פסוקיות היא $\sum p_i = 1$ ולכן

$$\frac{7m}{8} = E(X) = \sum_j p_j$$

וטענתינו היא ש־$\frac{c}{m}$ $\sum_{j\geq\frac{7m}{8}}p_j\geq$ ולכן

$$\begin{aligned}
\sum_{j=1}^{\frac{7m}{8}-1}jp_j+\sum_{j=\frac{7m}{8}}^{m}jp_j &= \sum_{j=0}^{m}jp_j=\frac{7m}{8}\\
u &= \sum_{j\geq\frac{7}{8}m}p_j\\
\left(\frac{7m}{8}-1\right)(1-u)+mu &\geq \frac{7m}{8}\\
\frac{7m}{8}-\frac{7m}{8}u-1+u+mu &\geq \frac{7m}{8}\\
\left(\frac{m}{8}+1\right)u &\geq 1\\
u &\geq \frac{1}{\frac{m}{8}+1}=\frac{8}{m+8}
\end{aligned}$$

■

6 טרנספורם פורייה דיסקרטי / מהיר - Discrete / Fast Fourier Transform

6.1 רקע

רקע: DFT זו אחת הפעולות הבסיסיות ביותר בתחום של עיבוד אותות (Signal Processing).
הרעיון הבסיסי טרנספורם פורייה: כל פונקציה מחזורית ניתן לבטא כצירוף של סינוסים וקוסינוסים:

$$f(x)\rightarrow\sum a_j\sin jx+\sum b_j\cos jx$$

נסתכל על f כך ש־$f:\mathbb{R}\rightarrow\mathbb{R}$ או $f:\mathbb{R}\rightarrow\mathbb{C}$ בעלת מחזור T, כלומר $\forall x, f(x+T)=f(x)$.
אנחנו נעבוד לא עם $f:\mathbb{R}\rightarrow\mathbb{C}$ אלא אם פונקציות $f:\{0,1,\ldots,n\}\rightarrow\mathbb{R},\mathbb{C}$, כלומר מבחינתינו f היא וקטור n מימדי.
גם לפי $x\rightarrow\sin kx$ ו־$x\rightarrow\cos lx$ נבצע דיסקרטיציה דומה.
בהקשר הדיסקרטי, הרעיון של פורייה מתממש בכך שאם הווקטור f המתאים לאיזושהי פונקציה) אנו נציג כצירוף של וקטורים מיוחדים (המתאימים לפונקציות $\sin kx,\cos kx$).
כלומר במרחב הוקטורים ה־n מימדיים יש איזשהם וקטורים מיוחדים ואנו רוצים להציג כווקטור כללי בעזרת צירוף לינארי שלהסן
מתברר שכאשר מציגים נכון את הדברים הוקטורים המיוחדים האלה מהווים בסיס אורתונורמלי למרחב ולכן ניתן באופן יעיל במיוחד למצוא את הייצוג של f הנתונה בצירוף של ווקטורים אלה.

תזכורת: אם $\vec{a},\vec{b}\in\mathbb{C}^n$ אז $\left\langle\vec{a},\vec{b}\right\rangle=\sum a_j\overline{b_j}$.
אומרים ש־$u_1,\ldots,u_n$ הם בסיס אורתונורמלי ל־$\mathbb{C}^n$ אם $\langle u_j,u_k\rangle=\delta_{jk}$ ....
ז"א $\langle u_j,u_k\rangle=0$ ולכל j מתקיים $||u_\triangle||=1$ יהיה אם כן $u_1,\ldots,u_n$ בסיס אורתונורמלי ל־$\mathbb{C}^n$ ויהיה $f\in\mathbb{C}^n$
נרצה להציג את f כצירוף לינארי של ה־u_i כש־$f=\sum\alpha_iu_i$ ו־$\alpha_i=\langle f,u_i\rangle$ מקבלים זאת ע"י הכפלה של השוויון עם u_j .

תכונה חשוב של הפונקציות הטריגונומטריות כאשר $k = 0, \dots, N-1$ כאשר $x \to \sin kx$ ו־$l = 0, \dots, N-1$ ו־$x \to \cos kx$ אז זהו בסיס אורתונורמלי.

יהי לנו נוח לעבוד עם וקטורים ב־$\mathbb{C}^n$ ולבטא אותם בבסיסי של הפונקציה

$$j = 0, \dots, N-1, v_k^{(j)} = e^{-2\pi ik/(N-1)}$$

כאשר j הוא המספר הסידורי של הוקטור , ו־k הוא הקואורדינטה ה־k של הוקטור.
הקשר לפי הטריגונומטריו הוא:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

בפרט, הקואורדינטה ה־k בווקטור ה־j שלנו, כלומר $u_k^{(j)}$ היא

$$\cos\frac{2\pi jk}{N} + i\sin\frac{2\pi jk}{N}$$

שורשי היחידה המרוכבים. למשוואה $z^N = 1$ יש N שורשים במישור המרוכב ולאלא הם $\omega = \omega_i = e^{2\pi i/N}$ וכל הפתרונות של המשוואה $z^0 = 1$ הם המספר המרוכב $1, \omega, \omega^2, \dots, \omega^N$.
בעזר הסימון הזה ניתן גם רשום את $u_k^{(i)} = \omega^{jk}$ נ"י

$$\begin{aligned} u^{(0)} &= (1, \dots, 1) \\ u^{(1)} &= 1, \omega\omega 2 \end{aligned}$$

המדוייק התכונה σ של הוקטורים $u^{(0)}, \dots, u^{(N-1)}$.

$$\left\langle u^{(m)}, u^{(s)} \right\rangle = \begin{cases} N & r = s \\ 0 & r \neq s \end{cases}$$

הוכחה: נזכיר שמכפלה פנימית בין וקטורים מעל $\mathbb{C}$ מוגדרת עוד $\left\langle \vec{a}, \vec{b} \right\rangle$

$$\left\langle u^{(k)}, u^6 \right\rangle = \sum_{k=1}^{n}$$

סיכום רגעי בבעיה. ואנחנו עובדים במרחב ה־N מימידי המרוכב.
מבטאים וקטור $osf = (f_0 \dots, f_{n+1}) \in \mathbb{C}^n$ כצירוף לינארי של וקטורים $u^{(0)}, \dots, , u^{(n+1)}$ כך ש־

$$\left\langle \overbrace{u}^{m}, v^7 \right\rangle = \begin{cases} 0 & r \neq z \\ N & \end{cases}$$

אם רוצים לבטא את $f = \sum_{j=0}^{n-1} \alpha_j u^{(j)}$ אז $\alpha_j = \frac{1}{N}\left(f, i^{(j)}\right)$.
טרנספורם פורייה דיסקרטי זו הפעולה שבה בהינתן $f \in \mathbb{C}^n$ אנו מציגים אותו כצירוף.
דרך שקולה לבטא פעולת DFT היא זו

$$f \mapsto \frac{1}{n}\begin{pmatrix} u^{(0)} \\ 0 \\ u^{(N-1)} \end{pmatrix}(p) = \begin{pmatrix} \alpha_0 \\ 0 \\ \alpha_n \end{pmatrix}$$

כאשר ω_{ij} היא מטריצה ה־DFT ופעולת הדפס מווגדרת ע"י

$$f \to (\omega^{pq})(f)$$

משפט 6.1 (ללא הוכחה) הפעולה בה יש לנו כקלט מטריצה $N \times N$, A וקטור מ־N מימדי y.
והפלט הוא A_y מחייבת את $\Omega\left(\mathbb{N}^2\right)$ פעולת אלגבריות.

סיכום: בהינתן N טבעי, מגדיירם את המטריצה של טרנספורם פורייה דיסקרטי ע"י $A_{N-1\times N-1}$ כאשר $a_{p,q} = \omega^{pq}$
ω הוא שורש היחידה הפרימיטיבי מסדר N, כלומר $\omega = e^{2\pi i/N}$.
DFT היא הפעולה $f \to Af$ ($f \in \mathbb{C}^n$) כפי שהבנו זו הפעולה המאפשרת לנו להציג אות מחזורי כצירוף מתאים של $\sin$ ו־$\cos$.
ראינו גם שהמטריצה A היא הפיכה וההופכית שלה $A^{-1} = \frac{1}{N}\left(\omega^{-pq}\right)$ ($A^{-1} = \overline{A}$).
כאשר f הוא האות בתחום הזמן (time domain), ו־$\hat{f} = Af$ הטרנספורם פורייה של f הוא האות בתחום התדר.

דוגמא: דוגמא לעיבוד של אות כשהו נתון בתחום התדר - בכל מדידה פיסיקלית צריך לקחת בחשבון את נוכחותו של רעש.
התופעה האופיינית היא שרעש נותן להופיע בתדירויות גבוהות. אם האות נתון לנו בתחום התדר\ כלומר אנו מכירים את Af, אז ניתן לסנן בקלות הת הערכים בתדר הגבוהה של האות.
מדדנו f ← מצאנו Af באמצעות DFT ← נוכל עתה לסנן את התדרים הגבוהים וכך למצוא h בלי הרעש ← חזרה לתחום התדר ע"י $A^{-1}h$.

גישה אחרת לסינון רעשים היא ע"י מוצוע בין שכנים - אם ידוע לנו שהאות האמיתי (ללא רעש) איננו אמור לעבור שינויים מהירים מאוד בזמן, אז ניתן לפעול כך: נגדיר פונקציה g $f \to g$ ע"י $g_k = \frac{1}{4}f_{k-1} + \frac{1}{2}f_k + \frac{1}{4}f_{k+1}$.

הגדרה 6.2 לפעולה הזו אנו קוראים קונבולוציה עם הפונקציה (או הוקטור) אם לכל i מתקיים $\left(0,\dots,\frac{1}{4},\overbrace{\frac{1}{2}}^{\text{i-th}},\frac{1}{4},0,\dots,0\right)$.

קונבולוציה של שני וקטורים f,g מסומנת ע"י $h = f * g$ מוגדרת כך

$$h_k = \sum_j f_j g_{k-j}$$

יש דמיון רב וקשר הדוק בין פעולות הקונבולוציה ופעולות הכפל של פולינומים

$$\begin{aligned}
P(x) &= \sum_{j=0}^{k} a_j x^j \\
Q(x) &= \sum_{j=0}^{l} b_j x^j \\
P(x)Q(x) = R(x) &= \sum_{j=0}^{k+l} c_j x^j \\
c_j &= \sum_i a_j b_{j-i}
\end{aligned}$$

עבור טרנספורם פורייה מתקיים $\widehat{f*g} = \hat{f}\hat{g}$.

6.2 DFT ופעולות על פולינומים

שתי הפעולות הבסיסיות עם פולינומים הוא חיבור וכפל.
אם P,Q שני פולינומים ממעלה N, אז סכומים ניתן לחשב בקלות בזמן $O\left(N\right)$ (מספר המקדמים המתאימים).
לעומת זאת, כפל פולינומים הוא לפחות באופן נאיבי מצריך זמן $\Omega\left(N^2\right)$.
נראה שניתן להשתמש ב-DFt ע"מ להכפיל פולינומים, כאשר אם נדע לממש את DFT בזמן $O\left(N\log N\right)$ (ע"י FFT)
אז נוכל באותו הזמן להכפיל שני פולינומים.

הייצוג הרגיל של פולינום הוא ע"י מקדמיו $P\left(x\right)=a_0+a_1x+\ldots+a_{N-1}x^{n-1}$.
יש שיטה אלטרנטיבית לייצוג של פולינומים. באופן כללי, פולינום P ממעלה $\geq d$ מוגדר ביחידות ע"י $d+1$ נקודות על הגרף שלו $\left(x_0,y_0\right),\ldots,\left(x_d,y_d\right)$ כאשר $P\left(x_i\right)=y_i$.

יחידות: תהיינה נתונות $d+1$ הנקודות $\left(x_j,y_j\right)_{j=0}^{d}$ (כאשר x_j שונים זה מזה) ונניח ש-P,Q הם פולינומים ממעלה $\geq d$ כך ש-

$$\forall_j P\left(x_j\right) = y_j, Q\left(x_j\right)=y_j$$

אז $R\left(x\right)=P\left(x\right)-Q\left(x\right)$ ולכן $\deg R\leq d$ וגם $\forall j,R\left(x_j\right)=0$ וכיוון ש-R הוא פולינום ממעלה $\geq d$ ולכן יש לו לפחות $d+1$ שורשים לכן $R=0$.

קיום: יש נוסחה מפורשת הנותנת את P, נוסחת האינטרפולציה של לגרנג'.
כיוון אחד הוא פשוט ע"י הצבת d ערכים.
הכיוון ההפוך נקרא אינטרפולציה.

כפל פולינומים: אם P,Q שני פולינומים ממעלה $\geq d$ כך ש-$d<\frac{N}{2}$ אשר מיוצגים ע"י N נקודות ו-$R=PQ$ אז $R\left(x_k\right)=P\left(x_k\right)Q\left(x_k\right)$ ולכן ניתן לכפול פולינומים בייצוג זה ב-$O\left(N\right)$, ובאותו אופן $T=P+Q$ אז $T\left(x_k\right)=P\left(x_k\right)+Q\left(x_k\right)$ ולכן גם החיבור הוא $O\left(N\right)$.

אם נבחר את $x_0,\ldots,x_{n-1}$ כשורשרי היחידיה הם ה-Nים אזי ניתן ע"י ה-DFT לעבור בין ייצוג ע"י מקדמים לייצוג ע"י ערכים בשורשי היחידה.
האלגוריתם FFT מאפשר לעשות את זה בזמן $O\left(N\log N\right)$.

היה $P\left(x\right)=a_0+a_1x+\ldots+a_dx^d$ פולינום, ואת ההצבה של הערך x_0בו ניתן לרשום ע"י $f\left(x_0\right)=\left\langle\left(a_0,a_1,\ldots,a_d\right),\left(1,x_0,x_0^2,\ldots,x_0^d\right)\right\rangle$.
אם נרצה את ערכי P בנקודות $x_0,\ldots,x_d$ אז נרשום את הפעולה במטריצה

$$\begin{pmatrix}1 & \ldots & x_0^d\\ \vdots & \ddots & \vdots\\ 1 & \ldots & x_d^d\end{pmatrix}\begin{pmatrix}a_0\\ \vdots\\ a_n\end{pmatrix}=\begin{pmatrix}P\left(x_0\right)\\ \vdots\\ P\left(X_d\right)\end{pmatrix}$$

וזו מטריצת ון דר מונדה, והראינו בלינארית 1 שעבור x_i שונים זה מזה, המטריצה אינה סינגולרית (כלומר הפיכה) ולכן ניתן באמצעות המטריצה וההופכית שלה לעבור בין הייצוגים השונים.

במידה ובחרנו את x_j כשורשי היחידה ה-Nים, כלומר $x_j=\omega^j$ ו-$\omega=e^{2\pi i/N}$ אז מטריצת ון דר מונגה הזו ניראת

$$\begin{pmatrix}1 & x_0 & \ldots & x_0^d\\ \vdots & & & \vdots\\ 1 & x_d & \ldots & x_d^d\end{pmatrix}=\begin{pmatrix}1 & 1 & \ldots & \\ 1 & \omega & \omega^2 & \ldots\\ \vdots & \ddots & \ddots & \ddots\end{pmatrix}$$

כלומר קיבלנו את מטריצת ה-DFT, ולכן הפעלת ה-DFT על וקטור המקדמים יתן את וקטור הערכים ובאופן דומה בכיוון ההפוך, ולכן אם ניתן לבצע את התהליך ב-$O\left(N\log N\right)$ אז ניתן להכפי פולינומים ממעלה N בזמן $O\left(N\log N\right)$.

6.3 אלגוריתם FFT

תזכורת: נגדיר את המטריצה $A_{pq}=\omega^{pq}$ כאשר $\omega=e^{2\pi i/N}$.

פעולה	בעיבוד אותות	בפולינומים
$(DFT)\ f\to Af$	מעבר מתחום הזמן לתחום התדר	מעריכים פולינום P לפי שורשי היחידה
$(DFT^{-1})\ g\to A^{-1}g$	מעבר מתחום התדר לתחום הזמן	אינטרפולציה - בהינתן ערכי הפולינום בשורשי היחידה נקבל את מקדמיו

איור 6.1: משמעות הפעולות של DFT וההופכית לו

כאשר תחום התדר הוא תיאור הפונקציה כצירוף של $\sin,\cos$ (בעצם כצירוף של אקספוננטים).
הגישה שלנו לכפל מהיר של פולינומים ע"י מקדמים היא ע"י מעבר מייצוג לפי פולינומים לייצוג לפי ערכים, שם ניתן להכפיל ב-$O(N)$ ולחזור חזרה לייצוג לפי מקדמים.

טרנספורם פורייה מהיר FFT ממש את פעולת ה-DFT בזמן $O(N\log N)$ (ולא בזמן N^2 כפי שהיה נדרש במימוש נאיבי של כפל מטריצה בווקטור).

דרך הפעולה: נתון לנו הפולינום $P(x)=a_0+a_1x+\ldots+a_{N-1}x^{N-1}$ ואנו מעוניינים בחישוב המספרים $P(\omega^k)$ לכל $k=0,\ldots,N-1$.
נניח בה"כ ש-N היא חזקה של 2, ונרצה למצוא אלגוריתם שהערכת זמן הריצה $T(N)$ תקיים באופן ריקורסיבי:

$$\begin{aligned} T(N) &\le 2T(N/2)+O(N)\\ &\Downarrow\\ T(N) &\le O(N\log N)\end{aligned}$$

כלומר נרצה לפתור את אותה הבעיה בעזרת שיטת "הפרד ומשול" עם שני פולינומים ממעלה $\frac{N}{2}$ ועוד עבודת עזר בזמן לינארי ב-N.

חלוקת הפולינום: אם נתון לנו פולינום

$$\begin{aligned} P(x) &= a_0+a_1x+a_2x^2+\ldots\\ &= \left(a_0+a_2x^2+a_4x^4+\ldots\right)\\ &\quad +x\left(a_1+a_3x^2+a_6x^4+\ldots\right)\\ E(z) &:= a_0+a_2z+a_4z^2+\ldots\\ D(z) &:= a_1+a_3z+a_6z^2+\ldots\\ \Rightarrow P(x) &= E\left(x^2\right)+xD\left(x^2\right)\end{aligned}$$

כאשר הפולינומים D,E הם ממעלה $\frac{N}{2}$.
ע"מ להעריך את P בכל שורשי היחידה ה-N-ים $\{P(\omega)\}_{k=0}^{n-1}$ די לנו :

1. להעריך את E בכל שורשי החידה מסדר $N/2$.
2. להעריך את D בכל שורשרי היחידה מסדר $\frac{N}{2}$.
3. להכפיל את D בשורש יחידה מתאים ולסכם.

אבחנה: אם מעלים שורש יחידה מסדר 2^s בריבועי מקבלים שורש יחידה מסדר 2^{s-1}.
מדוע? ניסתכל למשל על שורש יחידה מסדר 16 שנראה $\omega^j=e^{2\pi ij/16}$ ואז אם נעלה אותו בריבוע נקבל

$$\left(\omega^j\right)^2=\omega^{2j}=e^{4\pi j/16}=e^{2\pi i/8}$$

ובאופן כללי, עבור שורש יחידה מסדר 2^s, אז $\omega^j = e^{\frac{2\pi j}{2^s}}$ הוא שורש יחידה מסדר $\left(\omega^j\right)^2 = e^{\frac{2\pi j}{2^s}\cdot 2} = e^{\frac{2\pi j}{2^{s-1}}}$ הוא שורש יחידה מסדר 2^{s-1}.

נסכם: DFT מטרתיו להעריך פולינום נתון P ממעלה N (אנו מניחים כי היא בחזרה של 2).
אנו מציגים $P\left(x\right) = E\left(x^2\right) + xD\left(x^2\right)$, ולכן אם ω הוא שורש יחידה N־י ו־$N-1 \geq k \geq 1$ אז

$$P\left(\omega^k\right) = E\left(\left(\omega^k\right)^2\right) + \omega^k D\left(\left(\omega^k\right)^2\right)$$

אבל ω^{2k} הוא שורש יחידה מסדר $\frac{N}{2}$, ולכן אם נדע את כל ערכי E, D לפי שורשי יחידה מסדר $\frac{N}{2}$, ואז ע"י עוד $O\left(N\right)$ פעולות (ההכפלה של D ב־x) נקבל את מה שרצינו.
ההערכה של D בכל שורשרי היחידה מסדר $\frac{N}{2}$ היא DFT מסדר $\frac{N}{2}$, וכנ"ל עבור E, וכך קיבלנו את הריקורסיה שרשמנו בהתחלה, ולכן $T\left(N\right) \leq O\left(N\log N\right)$.

הרחבה על חיבור התוצאות: בשלב הראשון חישבנו את $D\left(\tau^k\right)_{k=0,\ldots,\frac{N}{2}-1}$ ו־$E\left(\tau^k\right)_{k=0,\ldots,\frac{N}{2}-1}$ כאשר $\tau = \omega^2 = e^{\frac{2\pi i}{\left(\frac{N}{2}\right)}}$.
כל מספר מהצורה $D\left(\tau^k\right)$ שחישבנו בריקורסיה, עלינו להכפיל עכשיו את ω^k, כאשר כ לפעולה כזאת היא הכפלה אחת של שני מספרים מרוכבים, ולכן בסה"כ $O\left(N\right)$ פעולות.

7 אלגברה לינארית חישובית

7.1 אלגוריתם Strassen לכפל מטריצות

ידוע שכפל מטריצה בוקטור (כשהן המטריצה והן הוקטור הם הקלט) מצריך $\Omega\left(n^2\right)$ פעולות אריתמטיות. האלגוריתם הנאיבי הסטנדרטי לכפל של שתי מטריצות $n \times n$ הוא $\Theta\left(n^3\right)$ פעולות אלמנטריות. ב־1973 הראה Strassen אלגוריתם לפתרות בעייה זו בסיבוכיות של $\Theta\left(n^{\log_2 7}\right) = \Theta\left(n^{2.83}\right)$. האלגוריתם הטוב ביותר הידוע כיום הוא $O\left(n^{2.36}\right)$.

הוא מצא אלגוריתם לכפל של מטריצות 2×2 המצריך רק 7 (ולא 8) פעולות כפל, אבל יתרונו בכך שהאלגוריתם הזה עובד גם אם החוק הקומוטטיבי אינו מתקיים.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$$

ולכן ניתן לנצל את האלגוריתם הזה ע"מ להכפל מטריצות $N \times N$ כדילקמן: עבור הכפל $A \cdot B$ נחלק כל מטריצה לבלוקים:

$$\begin{pmatrix} A_{11} & | & A_{12} \\ - & | & - \\ A_{21} & | & A_{22} \end{pmatrix}\begin{pmatrix} B_{11} & | & B_{12} \\ - & | & - \\ B_{21} & | & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & | & \ldots \\ - & | & - \\ \ldots & | & \ldots \end{pmatrix}$$

כלומר כפל המטריצה הוא כפל של הבלוקים לפי מטריצה 2×2, ולכן מתקבל החסם העליון הבא על הסיבוכיות של כפל מטריצות $N \times N$

$$\begin{aligned} T\left(N\right) &\leq \overbrace{7}^{\text{Multiplications}} \cdot T\left(\frac{N}{2}\right) + \overbrace{O\left(N^2\right)}^{\text{Additions}} \\ 7^{\log_2 N} &= N^{\log_2 7} \end{aligned}$$

(ניתן להראות את חישוב זמן הריצה ישירות ממשפט האב שנלמד בקורס מבני נתונים).

7.2 פתרון מערכת משוואות לא מדוייקות

אנחנו עוסקים בסיטואציה מעשית שבה אילו יכולנו לדעת את כל נתנוני הבעיה באופן מושלם, היה עלינו לפתור את המערכת הלינארית $Ax = b$. הקושי נובע מכך שאיברי A ו־b ידועים לנו רק בקירוב (למשלם הם תוצאות של מדידות). נניח ש־b הוא חד מימדי, רעיון מתבקש הוא לנסות להשיג עוד מידע על ה מטריצה A ע"י הוספת מדידות נוספות, כלומר הוספת משוואות נוספות שיתנו מטריצה חדשה $M \in M_{m\times n}(\mathbb{R})$ ווקטור גדול יותר $c \in \mathbb{R}^m$. עתה נצטרך לפתור את

$$(*) \quad Mx = b$$

כשהמטריצה M היא $m \times n$ כאשר $m > n$ וזוהי (כרגיל) מערכת יתירה over-determinnent כך שאין לצפות שיהיה למערכת $*$ פתרון שהו. כלומר שהדבר הטוב ביותר שאפשר לצפות לו הוא שנמצא x כך ש־Mx "קרוב" ל־b.
איך מודדים מרחקים בין וקטורים? כש־$u, v \in \mathbb{R}^m$ אז $d(u,v) = |u - v|$. הגדרות של מרחקים בין וקטורים מסתמכים על ההגדרה של אורך של וקטור (נורמה) $||\cdot||$. אם f, g הם וקטורים אז נירצה להגדיר $d(f,g) = ||f - g||$.

הגדרה 7.1 נרומה וקטורית היא העתקה $\mathbb{R}^n \to \mathbb{R}_+$ (כלומר למספרים אי שליליים) המקיימת:

1. חיוביות - לכל $v \in \mathbb{R}^n$ מתקיים ש־$||v|| \geq 0$.
2. הומוגניות - לכל $v \in \mathbb{R}^n$ ו־$a \in \mathbb{R}$ מתקיים ש־$||a \cdot v|| = |a| \cdot ||v||$.
3. א"ש המשולש - לכל $v, u \in \mathbb{R}^n$ מתקיים ש־$||u + v|| \leq ||u|| + ||v||$.

דוגמאות:

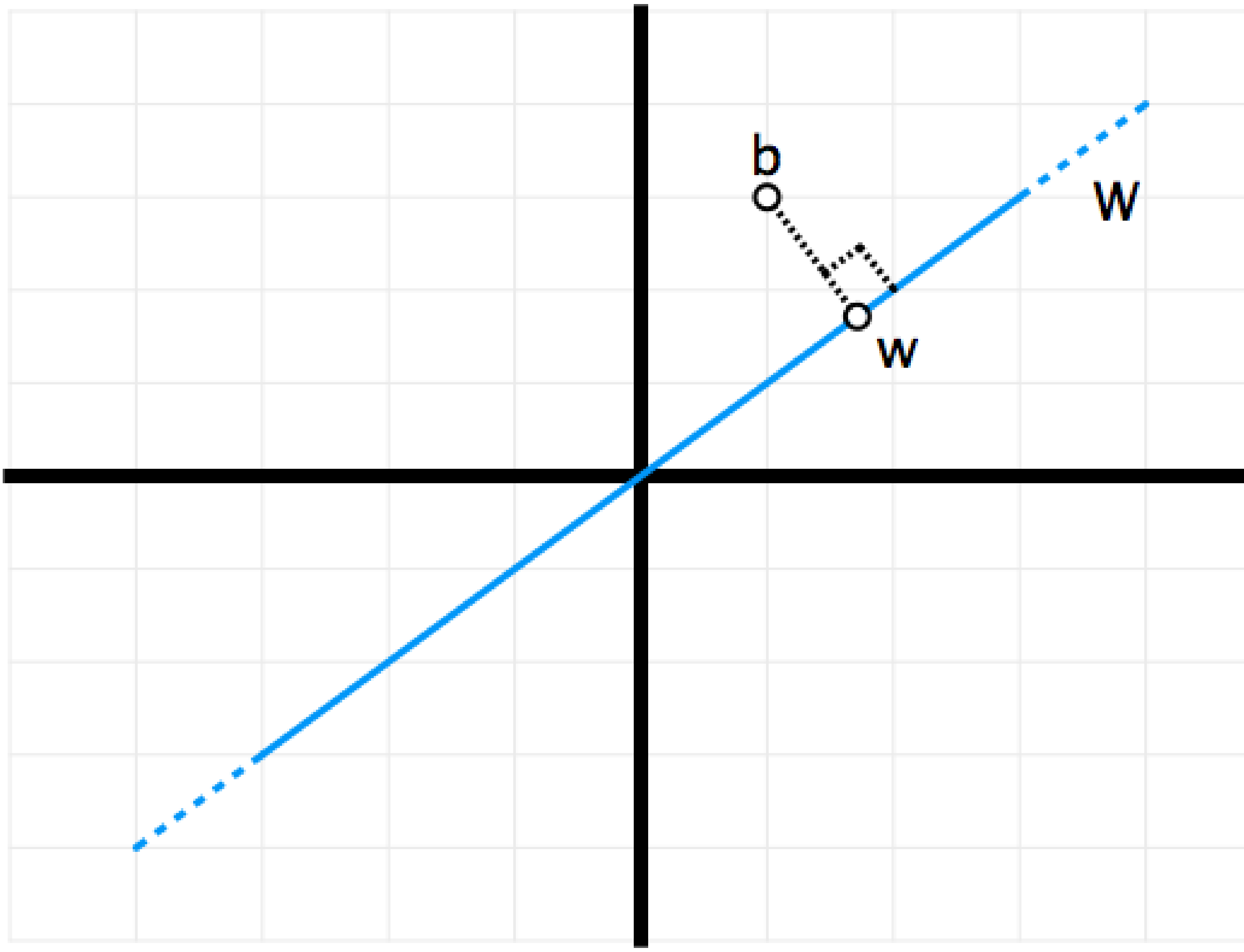
1. נורמת l_2 (נקראת גם נורמה אוקלידית) המוגדרת ע"י $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$

$$||u||_2 = \sqrt{\sum_{i=1}^{n} u_i^2}$$

2. נורמת l_1 מוגדרת ע"י $||u||_1 = \sum_{i=1}^{n} |u_i|$.
3. נורמת l_∞ מוגדרת ע"י $||u||_\infty = \max_{1\leq i\leq n} |u_i|$.

הבעיה: נתונה מטריצה $A_{m\times n}$ ו־ $b \in \mathbb{R}^m$ ומצא $x \in \mathbb{R}^n$ כך ש־$\min_{x\in\mathbb{R}^n} ||Ax - b||_2$, ואנו מניחים כי $rank(A) = n$ כלומר כל העמודות של A הן בת"ל.

נשים לב ש־$W = \{Az | z \in \mathbb{R}^m\}$ הוא תת מרחב n מימדי החלקי ל־$\mathbb{R}^m$, ולכן למעשה אנו מחפשים $w \in W$ כך ש־w קרוב ביותר (בנורמה l_2) ל־b, ולכן ב־$m = 2, n = 1$ מבחינה גיאומטרית ש־$b - w$ הוא ההיטל מ־b ל־W



איור 7.1: מציאת וקטור קרוב ביותר

הסיבה לכך שניתן לפתור ביעיליות את המקרה הזה היא ש־

$$\langle u,u\rangle = ||u||_2^2 = \sum_{i=1}^{n} u_i^2$$

ולכן

$$\begin{aligned} ||Ax-b||_2^2 &= \langle Ax-b, Ax-b\rangle \\ &= x^tA^tAx - 2b^rAx + ||b||_2^2 \end{aligned}$$

המעבר לעיל נובע מ־

$$\begin{aligned}\langle f,g\rangle &= f^Tg\\ (Ax-b)^t(Ax-b) &= \left(x^tA^t-b^t\right)(Ax-b)\\ &= x^tA^tAx-x^tA^tb-b^tAx+b^tb\\ &= x^tA^tAx-2b^tAx+||b||_2^2\end{aligned}$$

ונשים לב שמתקיים ש־$x^tMx=\sum_{i,j}m_{ij}x_ix_j$.

נמצא את $\min||Ax-b||_2$ ע"י גזירה והשוואה ל־0. נסמן $z:=A^tA$, אז כפי שראינו

$$\begin{aligned}||Ax-b||_2^2 &= x^tzx-2b^tAx+||b||^2\\ &= \sum_{ij}z_{ij}x_ix_j-\overbrace{2b^tAx}^{2\langle c,x\rangle}+||b||^2\\ c &:=\end{aligned}$$

נגזור לפי x_i ונשווה את הנגזרת לאפס:

$$\begin{aligned}\frac{\partial}{\partial x_i}\left(\langle c,x\rangle\right) &= \frac{\partial}{\partial x_i}\left(\sum_{i=1}^n c_ix_i\right)=c_i\\ \frac{\partial}{\partial x_i}\left(x^tzx-2b^tAx+||b||^2\right) &= 0\\ \Rightarrow\forall i,2\sum_{j=1}^n z_{ij}x_j &= 2\left(A^Tb\right)_i\\ \Rightarrow\forall i,(zx)_i &= \left(A^Tb\right)_i\end{aligned}$$

(ה־2 הוא כי המטריצה היא סימטרית, כלומר $z_{ij}=z_{ji}$). ולכן קיבלנו ש־

$$\begin{aligned}x &= z^{-1}A^Tb\\ &= \left(A^TA\right)^{-1}A^Tb\end{aligned}$$

למטריצה $\left(A^TA\right)^{-1}A^T$ קוראים ההפיך המוכלל של A.

מדוע יש ל־z הופכית? דבר ראשון נשים לב ש־A^TA היא $n\times n$ והיא מדרגה n ולכן הפיכה.

7.3 קירוב ע"י עקומות

ראינו פתרון לבעיית האינטרפולציה: בהינתן $(x_0,y_0),\ldots,(x_k,y_k)$ אז יש פולינום יחיד P ממעלה $\geq k$ כך ש־$p(x_i)=y_i$.

מה ניתן לעשות אם נתון לנו $(x_1,y_1),\ldots,(x_n,y_n)$ כש־$n>k+1$, אז לא ניתן לצפות שיהיה פולינום Q ממעל $k\geq$ המקיים $Q(x_i)=y_i$ לכל i.

לכן נרצה למצוא בהינתן $\{(x_i,y_i)\}_{i=1}^n$, פולינום R ממעלה $\geq k$ כך ש־$R(a_i)\simeq y_i$, כלומר $\min||(R(x_1),\ldots,R(x_n))-(y_1,\ldots,y_n)||_2$.

נוכל לתרגם את הבעייה למטריצה

$$\begin{pmatrix}1 & x_1 & \ldots & x_1^k\\ \vdots & \ddots & & \vdots\\ 1 & x_n & \ldots & x_n^k\end{pmatrix}\begin{pmatrix}a_0\\ \vdots\\ a_k\end{pmatrix}=\begin{pmatrix}R(x_1)\\ \vdots\\ R(x_n)\end{pmatrix}$$

ולכן הבעייה של אינטרפולציה מקורבת היא מקרה פרטי של פתרון מקורב למערכת לינארית יתירה.

7.4 חזרה לבעיית מערכת המשוואות הלינאריות

נחזור ונתבונן במצב שבו עלינו לפתור מערכת משוואות לינאריות $Ax = b$. לצורך הדיון הנוכיח אנחנו איננו מטילים ספר המטריצה A אבל הוקטור b "מפוקפק".

נניח שאגף ימין האמיתי איננו b אלאל $b + \Delta b$. נניח שאת $Ax = b$ פתרנו, ואז מתקיים $A(x + \Delta x) = b + \Delta b$.
המצב שממנו אנו רוצים להימנע הוא ש-$\frac{||\Delta b||}{||b||}$ קטן, אבל $\frac{||\Delta x||}{||x||}$ גדול.

השאלה שבה נתעניין היא זו: מהם התנאים ש-A צריכה לקיים ע"מ שטעות יחסית קטנה בידיעת הוקטור b (כלומר $\frac{||\Delta b||}{||b||}$ קטנה) תתבטא רק בטעונות קטנה בפתרון (כלומר $\frac{||\Delta x||}{|m|x|}$ קטן).

על יציבות נומרית: במחשב מספרים מיוצגים בייצוג סופי, ולכן מספרים אי רציונליים (ואפילו מספרים רציונליים מסויימים) לא ניתנים לייצוג במחשב במאופן מפורש - ולכן גם אם ידוע הערך המדוייק של מספר, בגלל אופן ייצוג המספר במחשב, למעשה קיים אי דיוק.

דוגמא: אם נסתכל למשל על האלגוריתם לדירוג מטריצה לפי גאוס (אלימינטציית גאוס – זהו האלגוריתם הסטדנדרטי עם פעולות על שורות), אז זאת דוגמא לאלגוריתם שהוא לא יציב נומרית.
נזכיר את המשפט - לכל מטריצה $A_{n\times n}$ לא סינגולרית, ניתן למצוא מטריצה משולשת עליונה U ותחתונה L ומטריצת תמורה P (מטריצה שבכל עמודה ובכל שורה יש איבר יחיד בעל ערך 1) כך ש-$PA = LU$ ולכן $A = P^T LU$ ונשים לב ש-$P^T = P^{-1}$ (כי P אורתוגונלית), ולכן כיוון ש-$P^T LUx = Ax = b$ אז $LUx = Pb$, כלומר זהו סידור מחדש של הקואורדינטות של b.
עתה נוכל להגדיר $Ux = y$ ואז $Ly = Pb$, וזו מערכת שאפשר לפתור ע"י חילוק וחיסור בודדים בכל משוואה (כי במשוואה הראשונה היא מהצורה $ax = b$, השנייה היא $a'x + cy = b'$ אבל את x אנחנו כבר יודעים, וכך הלאה).

שאלה: האם בהינתן ש-$\frac{||\Delta b||}{||b||}$ קטנה, האם ניתן גם להוכיח ש-$\frac{||\Delta x||}{||x||}$ קטנה.

אם התשובה חיובית, אז אומרים ש-A מיוצגת היטב (well-posed) ואחרת ill-posed.

הקושה מתעורר למשל אם A "כמעט סינגולרית" כלומר $A = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1+\epsilon & 1-\epsilon \end{pmatrix}$ ואז $A^{-1} = \frac{1}{\epsilon}\begin{pmatrix} -1+\epsilon & 1 \\ 1+\epsilon & -1 \end{pmatrix}$.

איך מודדים גודל של מטריצה?

1. ניתן לחשוב על מטריצה $m \times n$ כוקטור mn ואז $||A|| = \sqrt{\sum a_{ij}^2}$.
2. קיימת הגדרה הנקראה נורמה אופרטורית המודדת את ה"מתיחה" המירבית ש-A מבצעת על וקטור x, ולכן נגדיר

$$||A||_{op} = \max_{x\in V} \frac{||Ax||_W}{||x||_V}$$

נציג מספר תכונות של הנורמה האופרטורית:

(א) לכל $y \in V$ מתקיים $||Ay|| \le ||A||_{op} \cdot ||y||$.
(ב) $||A||_{op}\, ||B||_{op} \ge ||AB||_{op}$.

באילו תנאים נוכל לאמר ש-$\frac{||\Delta x||}{||x||}$ קטן במונחי $\frac{||\Delta b||}{||b||}$? ניתן תניא מספיק –

$$\begin{aligned}
||A||\,||x|| &\ge ||Ax|| = ||b|| \\
\left|\left|A^{-1}\right|\right|\,||\Delta b|| &\ge \left|\left|A^{-1}\Delta b\right|\right| = ||\Delta x|| \\
\Rightarrow ||A||\,\left|\left|A^{-1}\right|\right|\,||x||\,||\Delta b|| &\ge ||b||\,||\Delta x|| \\
\overbrace{||A||\,\left|\left|A^{-1}\right|\right|}^{\kappa_A}\frac{||\Delta b||}{||b||} &\ge \frac{||\Delta x||}{||x||}
\end{aligned}$$

כאשר κ_A נקרא conditional number.

הערה 7.2 אם נציב $B = A^{-1}$ נקבל בתכונה השנייה כי $1 = ||I|| \le \left|\left|AA^{-1}\right|\right| \le ||A||_{op} \left|\left|A^{-1}\right|\right|$.

7.4.1 משפט ה־SVD – Singular Value Decomposition – פירוק ערכים סינגולרים

הגדרה 7.3 אומרים ש־$A_{n\times m}$ מטריצה אורתוגונלית אם $AA^T = I$.
הגדרה שקולה – השורות של A מהוות בסיס אורתונורמלי למרחב ה־n מימדי.

הגדרה 7.4 הפעולה של מטריצה אורתוגונלית על $\mathbb{R}^n$ היא איזומטרית, כלומר כל הוקטורים שומרים על אורכיהם, והזוויות בינהן.

הערה 7.5 נשים לב שמתקיים עבור A אורתוגונלית כי

$$||Ax||_2^2 = \langle Ax, Ax\rangle = x^T \overbrace{A^T A}^{I} x$$

משפט 7.6 כל ט"ל $R^m \to \mathbb{R}^n$ ניתן לממש שהרכבה של שלוש פעולות:

1. איזומטריה על $\mathbb{R}^m$.
2. הטלה ל־n הקואורדינטות הראשונות והכפלה בקבועים $(u_1, \ldots, u_m) \to \{\alpha_1 u_1, \ldots, \alpha_m u_m\}$.
3. איזומטריה על $\mathbb{R}^n$.

משפט 7.7 (משפט ה־SVD) $A_{m\times n}$ כאשר $m \ge n$. לכל מטריצה $A_{m\times n}$ כש־$n \ge m$ יש הצגה $A = U_{n\times m} D_{m\times n} V_{n\times m}^T$ כש־U, V מטריצות אורתוגונליות ו־D מטריצה אלכסונית כשאיברי האלכסון הם $0 \ge \sigma_n \ge \ldots \ge \sigma_1$. המספרים $\sigma_1, \ldots, \sigma_n$ נקבעים ביחידות ונקראים הערכים הסינגולריים של A.

תכונות טובות של פירוק SVD:

1. $||A||_{op} = \sigma_1$.
2. נניח שנתונה A ונתון מספר טבעי k, ומחפשים מטריצה B בדרגה $k \ge$ המקרבת את A בצורה הטובה ביותר, במושב ש־$\min ||A - B||_{op}$ / $\min ||A - B||_{l_2}$.

משפט 7.8 אם $A = UDV^T$ כאשר D אלכסונים עם σ_i על האלכסון, אז B האופטימלית (הן במובן של נורמה אופרטורים והן במובן של l_2 היא $UD^{(k)}V^T$.

8 בעיות בתורת המספרים

8.1 מספרים ראשוניים

כל מספר ראשוני ניתן לכתיבה ע"י $n = \prod_{i=1}^{\infty} p_i^{e_i}$ כאשר p_i ראשוני ו־$e_i \in \{0\} \bigcup \mathbb{N}$.
בעיות טבעיות המתעוררת כאן:

8.1.1 מבחן ראשוניות

בהינתן לנו מספר טבעי איך ניתן להכרעי אם הוא ראשוני או לא (מבחן ראשוניות).
כשהקלט הוא המספר הטבעי n, אז זמן פולינומי פירושו $O\left((\log n)^c\right)$. כש־c קבוע כלשהו.

משפט 8.1 (משפט פרמה הקטן) אם p ראשוני ו־$1 \leq a \leq p-1$ אזי $a^{p-1} \equiv 1 \pmod p$.

נראה כאן אלגוריתם הסתברותי למבחן ראשוניות, ונניח שהקלט למבחן הראשוניות הם המספר n, ונניח שמצאנו איזשהו $1 \leq a \leq n-1$ כך ש־$a^{n-1} \not\equiv 1 \pmod n$ אז n אינ ראשוני, ונאמר ש־a הוא עד (witness) לפריקות של n.

8.1.2 פירוק לגורמים ראשוניים

בהינתן n טבעי, נרצה למצוא את ההצגה של n כמכפלה של חזקות ראשוניים. לבעיה זו לא מוכר אלגוריתם פולינומי.

8.1.3 כמה מספרים ראשוניים יש בין 1 ל־N

התשובה לבעיה זו נקראת משפט המספרים הראשוניים ומתקיים כי $\pi(N) = (1 + o(n)) \frac{N}{\log N}$

8.1.4 השערת רימן

השערת רימן אומרת שלכל $\epsilon > 0$ ול־N מספיק גדול יש מספר ראשוני בקטע שבין N ל־$N + N^{\frac{1}{2}+\epsilon}$.

8.2 רקע על תורת החבורות

הגדרה 8.2 חבורה $(G, \cdot)$ היא קבוצה G עם פעולה $\cdot$ המקיימת את התוכונות הבאות:

1. לכל $a, b \in G$ הפעולה $a \cdot b$ מוגדרת ו־$a \cdot b \in G$.
2. אסוציאטיביות: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
3. קיום איבר היחידה: יש אביר $e \in G$ כך שלכל $x \in G$ מתקיים $e \cdot x = x \cdot e = x$.
4. קיום איבר הופכי: לכל $g \in G$ קיים $h \in G$ כך ש־$hg = gh = e$.

דוגמאות:

1. חיבור מודולו n הוא חבורה $G = \{0, 1, \ldots, n-1\}$ עם פעולת החיבור ואיבר האפס הוא איבר היחידה.
2. כפל מודולו p ראשוני $G = \{1, \ldots, p-1\}$ כאשר איבר היחידה הוא 1.

הערה 8.3 שתי הדוגמאות הנ"ל הן חבורות קומוטטיביות או אבליות, כלומר בהן מתקיים שלכל $x, y \in G$ מתקיים כי $xy = yx$, אבל לא כל החבורות הן אבליות.

יש תיאור מלא של כל החבורות האבליות הסופיות לפי "משפט המיון של חבורת אבליות סופיות".

דוגמא: חבורה סופית לא קומוטטיבית היא למשל S_n, חבורת התמורות על $\{1, \ldots, n\}$, ומתקיים ש־$|S_n| = n!$, והפעולה היא פעולה ההרכבה, ואיבר היחידה היא תמורת הזהות.

האם ניתן לבנות חבורה כפלית מודולו n? אם $G=\{1,\ldots,n\}$ אז לא ניתן כי $4\cdot 3=0 \pmod n$ ולכן בשביל לבנות חבורה כזו, נרצה להשאיר ב־G רק את מספרים שזרים ל־n (תזכורת: אומרים ששני מספרים טבעים x,y הם זרים אם $gcd\left(x,y\right)=1$, כלומר אין להם מכנה משותף מלבד 1).

בעיה: מהו מספר האיברים ב־$\mathbb{Z}_n^*$? כלומר כמה מספרים בתחוסר $1,\ldots,n-1$ זרים ל־n?

תשובה: $\varphi\left(n\right)$ של אויילר המקיימת

$$\varphi\left(n\right)=n\prod_{\substack{p_i \text{ is prime}\\ p_i\mid n}}\left(1-\frac{1}{p_i}\right)$$

למשל $\varphi\left(30\right)=30\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{5}\right)=8$

חלק II

תרגולים

9 תרגול 1 – 12.10.2010

9.1 אדמיניסטרציה

אתר הקורס: http://www.cs.huji.ac.il/~algo

אי מייל הקורס: algo@cs.huji.ac.il

תרגילים: כל שבוע יפורסם תרגיל, אך לא צריך להגישם, אלא להביא אותם לראיונות. בכל ראיון יועלו 2 התרגילים האחרונים ויש להביא לראיון עותק של התרגיל. בראיון ייבדק גם התרגיל עצמו וגם ההצגה של הפתרונות.

הציון הסופי: 20% ראיונות ו־80% המבחן הסופי.

9.2 תזכורות

9.2.1 נוטוציות אסימפטוטיות

נניח שקיימת פונקציה $g:\mathbb{N}\to\mathbb{R}^+$.

הגדרה 9.1 $f=O\left(g\right)$ אם"ם קיים קבוע c כך ש־$f\le c\cdot g$.

הגדרה 9.2 $f=\Omega\left(g\right)$ אם"ם קיים קבוע c כך ש־$f\ge c\cdot g$.

הגדרה 9.3 $f=\Theta\left(g\right)$ אם"ם $f=O\left(g\right)$ וגם $f=\Omega\left(g\right)$.

הגדרה 9.4 $f=o\left(g\right)$ אם"ם $\lim\frac{f}{g}=0$.

הגדרה 9.5 $f=\omega\left(g\right)$ אם"ם $\lim\frac{g}{f}=0$.

9.2.2 גרפים

הגדרה 9.6 גרף הוא הזוג $G = \langle V, E \rangle$ כך ש־V היא קבוצת הקודקודים של הגרף ו־E היא קבוצה של זוגות (סדורים או לא סדורים אם הגרף מכוון או לא) המייצגים אץ הקשתות בין כל שני קודקודים.

הגדרה 9.7 שני קודקודים בעלי צלע משותפת נקראים שכנים.

הגדרה 9.8 מסילה בגרף היא סדרת קודקודים כך שבין כל שני קודקודים עוקבים בסדרה הם שכנים.

הגדרה 9.9 גרף קשיר הוא גרף בו קיימת מסילה בין כל שני קודקודים בגרף.

הגדרה 9.10 רכיבי קשירות היא תת קבוצה מקסימלית של קודקודים המהווים גרף קשיר.

הגדרה 9.11 מעגל בגרף הוא מסילה המתחילה ונגמרת באותו קודקוד. מעגל פשוט הוא מעגל המתחיל ונגמר באותו קודקוד שלו עובר באותו קודקוד פעמיים.

הגדרה 9.12 עץ הוא גרף קשיר חסר מעגלים.

תכונות על עצים:

1. בעץ על n קודקודים יש בדיוק $n-1$ צלעות.
2. הגדרות שקולות לעץ:
   (א) גרף קשיר שאי אפשר להחסיר ממנו צלע ולהישאר קשיר.
   (ב) גרף חסר מעגלים שאי אפשר להוסיף לו צלע.
3. אם u, v קודקודים בעץ, יש ביניהם מסילה יחידה.
4. הוספת צלע בעץ יוצרת מעגל יחיד.

9.3 אלגוריתם חמדן ל־MST

הגדרה 9.13 עץ פורש הוא עץ המהווה תת־גרף של G, המכיל את כל הקודקודים של העץ.

הגדרה 9.14 עץ פורש מינימלי (MST - Minimum Spanning Tree) : $G = \langle V, E \rangle$ ופונקציית משקל $\omega : \vec{E} \to \mathbb{R}$. יהי T עץ פורש של G, אז $\omega(T) = \sum_{e \in T} \omega(e)$. עץ פורש מינימלי אם המשקל שלו קטן או שווה למשלק של כל עץ פורש אחר.

9.3.1 אלגוריתם קרוסקל למציאת MST

האלגוריתם:

1. מיין את הצלעות לפי סדר עולה של משקולות.
2. נניח שהצלעות הן $(e_1, \ldots, e_n)$. נאתחל את העץ הפורש להיות $T = \langle V, \emptyset \rangle$ (בלי צלעות).
3. עוברים על הצלעות לפי הסדר, מהמשקל הנמוך לגבוהה, ואם צלע e סוגרת מעגל ב־T (ב־$T \cup \{e\}$ יש מעגל), אז לא מוסיפים אותה ל־T, אחרת כן.

נכונות האלגוריתם:

1. נראה ש־T הוא עץ. הוכחה: ב־T אין מעגלים (ברור), בנוסף T הוא קשיר ונוכיח בשלילה.
נניח שי ב־T כמה רכיבי קשירות $c_1, \ldots, c_k$. G הוא קשיר ולכן יש c_i ו־c_j שיש בינהן צלע ב־G.
הוספת הצלע הזו ל־T לא תיצור מעגל כי אין בינהם מסילה לפני ההוספה, ולכן האלגוריתם היה מוסיף את הצלע לעץ בסתירה.

2. נראה שהמשקל של T הוא מינימלי. נעזר באינדוקציה:

(א) טענת האינדוקציה: נניח ש־$(e_1, \ldots, e_{n-1})$ הן הצלעות של T. לכל $0 \le k \le n-1$ יש עץ פורש מינימלי M_k המתלכד עם הרישא של T כך ש־$e_1, \ldots, e_k$ הן צלעות ב־M_k.

(ב) בסיס האינדוקציה: אם $k = 0$ אז בכל עץ פורש מינימלי מתלכד עם $\emptyset$.

(ג) מעבר: יהי M_k עץ פורש מינימלי כך ש־$(e_1, \ldots, e_k)$ הן צלעות ב־M_k
נסמן $e = e_{k+1}$. אם $e \in M_k$, אז סיימנו ו־M_{k+1} מכיל את $(e_1, \ldots, e_{k+1})$.
אחרת, $e \notin M_k$ ולכן הוספת הצלע ל־M_k יוצר מעגל יחיד C.
יש צלע f ב־G שאינו ב־T, כי אחרת היה ב־T מעגל.

(ד) נסכם: $M_k \not\ni e \in T$, $M_k \ni f \notin T$, וב־$M_k \cup \{e\}$ יש מעגל יחיד C. נגדיר $M_{k+1} = (M_k \setminus \{f\}) \cup \{e\}$.

(ה) M_{k+1} (הנ"ל) הוא עץ.
הוכחה: יש ב־M_{k+1} $n-1$ צלעות ואין בו מעגלים כי ב־$M_k \cup \{e\}$ יש מעגל יחיד C ש־f הוא צלע בו ולכן אחרי החסרת f, מחסירים גם את המעגל C מהגרף.

(ו) M_{k+1} הוא בעל משקל מינימלי.
הוכחה: נראה תחילה ש־$\omega(e) \le \omega(f)$. נניח שלא, אז $\omega(f) < \omega(e)$, אז האלגוריתם החמדן ראה את f לפני שהוא ראה את e.
$f \notin T$ ולכן f יצר מעגל עם הצלעות שכבר היו ב־T.
הצלעות שכבר היו ב־T הן $e_1, \ldots, e_l$, $l \le k$, אבל $e_1, \ldots, e_k \in M_k$, ולכן קבוצת הצלעות $\{e_1, \ldots, e_l, f\}$ יש בה מעגל, ו־ $f \in M_k$ בסתירה.
נראה ש־

$$
\begin{aligned}
\omega(M_{k+1}) &= \sum_{g \in M_{k+1}} \omega(g) \\
&= \sum_{g \in M_k} \omega(g) + \overbrace{\omega(e) - \omega(f)}^{\le 0} \\
&\le \sum_{g \in M_k} \omega(g) = \omega(M_k)
\end{aligned}
$$

9.4 בעיית מיכל הדלק

קלט: גודל המיכל $x > 0$ (מלא בתחילת הדרך), סדרת המרחקים של התחנות $a_1, \ldots, a_n$.

פלט: $b_1, \ldots, b_m$ $(m \le n)$ התחנות שעוצרים בהם, כך ש:

1. $x \ge b_{i+1} - b_i$
2. m מינימלי.

הנחה: נניח כי קיים אלגוריתם כזה (פשוט לשנות את האלגוריתם כך שיבדוק אם ניתן להגיע ליעד).

האלגוריתם: בכל שלב נוסעים לתחנה הרחוקה ביותר שאפשר להגיע אליה.

הוכחת נכונות: נניח בשלילה שקיים שהפתרון לא אופטימלי. הפתרון של האלגוריתם החמדן יסומן ב־$B = (b_1, \dots, b_m)$.

יהי C, הפתרון האופטימלי שמסכים עם B על רישא מקסימלית של תחנות.
כלומר קיים k כך ש־$b_1, \dots, b_k \in C$ ולכל פתרון אופטמלי אחר C', $b_1, \dots, b_{k+1}$ לא כולם שייכים ל־C'.
נסתכל ב־C, $b_1, \dots, b_k, c_{k+1}, \dots, c_l$ כאשר $l < m$, ונחליף את c_{k+1} ב־b_{k+1}.
b_{k+1} נבחר להיות הרחוק ביותר מ־b_k שאפשר להגיע אליו, ולכן $b_{k+1} \le c_{k+1}$. ידוע כי $x \ge c_{k+2} - c_{k+1}$ ולכן $x \ge c_{k+2} - b_{k+1}$ ולכן הפתרון החדש חוקי. יש בו l תחנות ולכן הוא אופטימלי.

בנוסף, $b_1, \dots, b_{k+1}$ שייכים לפתרון האופטימלי החדש בסתירה למקסימליות של k.

10 תרגול 2 – 19.10.2010

10.1 דוגמא נוספת לגרף חיתוכים

בעיה: בקורס יש מספר זוגי של סטודנטים, וצריך להגיש תרגילים בזוגות.
כל סטודנטי גר בנקודה (x_i, y_i) על המישור, ומוכן ללכת לכל היותר r_i ק"מ ע"מ להיפגש עם בן/בת זוגו.
איך מזווגים אותם?

פתרון: נבנה גרף שבו קודקוד לכל סטודנט, ויש צלע בין כל 2 סטודנטים שמעגליהם נחתכים.
(זיווג מושלם: חלוקה של הקודקודים לזוגות כך שבין כל 2 קודקודים בזוג יש צלע.
קיימים אלגוריתמים יעלים שמוצאים זיווג מושלם בגרף במידה וקיים כזה)

10.2 בעיית Knapsack

בעיה: גנב נכנס לחנות עם n חפצים. לכל חפץ מחיר ומשקל: $(w_1, p_1), \dots, (w_n, p_n)$.
הגנב צריך לבחור תת קבוצה של $I \subset \{1, \dots, n\}$ כך ש:

1. $\sum_{i \in I} w_i \le W \in \mathbb{R}^+$
2. $\sum_{i \in I} p_i$ מקסימלי.

בעיה זו היא NP־שלמה ולכן לא נעסוק בפתרונה בקורס זה.

10.3 מה זה NP־שלם?

- מחלקת הבעיות שיש עבורן אלגוריתם פולינומי נקראת P.
- NP היא מחלקה של בעיות שמכילה את P וגם מכילה בעיות שמעריכים שאין להן אלגוריתם פולינומי.
- בעיות NP־שלמות ־ היא קבוצת בעיות שניתן לתרגם כל בעיה אחרת אליהן. אם יש לאחת מהן אלגוריתם פולינומי, אז לכל הבעיות ב־NP יש פתרון.
- לכן, זה יהיה מאוד מפתיע אם ימצא אלגוריתם פולינומי לבעיה NP־שלמה.

10.4 בעיית Knapsack שברי

בעיה: כמו הבעיה הרגילה אבל ניתן לקחת חלק מחפץ $(w_1,q_1),\ldots,(w_n,q_n)$, ו־$W\in\mathbb{R}^+$.

פלט: נייצג את הפתרון בוקטור fl $(t_1,\ldots,t_n)$כך ש:

- $0\le t_i\le w_i$
- $\sum_{i=1}^n t_i\le W$
- $\sum_{i=1}^n t_iq_i$ מקסימלי.

פתרון: נעזר באלגוריתם חמדן:

- מיין את החפצים כך ש־$q_1\ge q_2\ge\ldots\ge q_n$
- כל עוד נשאר לגנב מקום בתיק, הוא ימלא את התיק מהמוצר היקר לזול ביותר.

נכונות:

- נתגרם את הבעיה לבעיה אחרת ־ מצא $t=(t_1,\ldots,t_n)$שמקיים:
  - $0\le t_i\le w_i$
  - $\sum_i t_i\le W$
  - $\sum q_it_i$ מקסימלי (עובדה: פונקציה רציפה מקבוצה סגורה וחסומה מקבלת מקסימום, ולכן הוא קיים בבעיה זו).
- הוכחת האופטימליות של האלגוריתם:
  - נראה שאם יש פתרון אופטימלי שמסכים עם הפתרון החמדן על $k-1$ חפצים, אז יש פתרון אופטימלי שמסכים עם החמדן על k חפצים.
  - סימונים:
    * נסמן את הפתרון החמדן ב־$(x_1,\ldots,x_n)$.
    * נסמן את הפתרון האופטימלי ב־$(y_1,\ldots,y_n)$.
  - הנחת האינדוקציה: $x_1=y_1,\ldots,x_{k-1}=y_{k-1}$.
  - נניח שהחפצים ממוינים לפי המחיר לק"ג (כמו באלגוריתם).
  - נשים לב שמתקיים:
    * $\sum x_i=\sum y_i=W$.
    * $y_k\le x_k$ ־ כי x_k נבחר ע"י החמדן, ולכן לא ניתן למלא את השק ביותר מ־x_k.
  - אם $x_k=y_k$ אז סיימנו, אחרת:
    * נסמן $0<s=x_k-y_k$, $r=y_{k+1}+\ldots+y_n$.
    * טענה: $r\ge s$.
      הוכחה:

$$\begin{aligned}y_{k+1}+\ldots+y_n &= W-(y_1+\ldots+y_k)=W-(x_1+\ldots+x_{k-1}+x_k-s)\\ &= \overbrace{W-(x_1+\ldots+x_k)}^{\ge 0}+s\ge s\end{aligned}$$

    * נסמן $0<\alpha=\frac{s}{r}\le 1$. עבור $k+1\le i\le n$ $y_i'=\alpha y_i$. הערות:
      - $\sum_{i=k+1}^n y_i'=\alpha\sum_{i=k+1}^n y_i=\alpha\cdot r=\frac{s}{r}\cdot r=s$

· $y'_i \le y_i$

* נגדיר את הפתרון האופטימלי החדש $Opt' = \left(y_1, \ldots, y_{k+1}, y_k + s, y_{k+1} - y'_{k+1}, \ldots, y_n - y'_n\right)$.

· הפתרון Opt' חוקי:

$$\begin{aligned} cost\left(Opt'\right) &= \sum_{i=1}^{n} y_i q_i + s \cdot q_k - \sum_{i=k+1}^{n} y'_i q_i \\ &= cost\left(Opt\right) + s \cdot q_k - \sum_{i=k+1}^{n} y'_i q_i \\ &\ge cost\left(Opt\right) + s \cdot q_k - \sum_{i=k+1}^{n} y'_i q_k \\ &= cost\left(Opt\right) + s \cdot q_k - q_k \cdot s = cost\left(Opt\right) \end{aligned}$$

10.5 מטרואידים

הגדרה 10.1 מטרואיד הוא זוג (S, I) כאשר S היא קבוצה סופית ו־$I \neq \emptyset$ קבוצה של תתי קבוצות של S (כלומר $I \subset 2^S$) המקיימת:

- תורשתיות: $A \in I$ ו־$B \subset A$ אזי $B \in I$.
- תכונת ההחלפה: $A, B \in I$ וגם $|B| > |A|$, אזי קיים איבר $a \in A \backslash B$ כך ש־$B \bigcup \{a\} \in I$.

דוגמא: מטראויד גרפי ־ יהי $G = \langle V, E \rangle$ גרף ונגדיר $S = E$. $I = \{A \subseteq E | \, (V, A) \text{ is a graph with out cycles}\}$.
נראה שזהו מטרואיד:

- אם ב־A אין מעגלים אז גם ב־ $B \subseteq A$ אין מעגלים.
- ב־A, B אין מעגלים, ונניח $|A| > |B|$, אז ב־B יש יותר רכיבי קשירות מב־A, ולכן יש צלע ב־A המחברת בין שני רכיבי קשירות שונים ב־B.

בעיה: נניח שרוצים למצוא את האיבר של I עם המשקל המקסימלי.

פתרון: נמיין את אברי S לפי משקלם, ונעבור על איברי S לפי הסדר, ובכל צעד נצרף את האיבר כל עוד הקבוצה שייכת ל־I.
בסוף התהליך הקבוצה שמתקבלת היא המועמד לקבוצה עם המשקל המקסימלי.

11 תרגול 3 – 26.10.2010

נציג בתרגול מספר דוגמאות לשימוש בתכנון דינמי.

11.1 בעיית ניתוב המשימות

בעיה: נתון גרף בעל נקודת התחלה s וסוף t, ואנו מחפשים מסלול בין s ל־t בעל משקל מינימלי בגרף זה (משקל המסלול הוא סכום המשקולות של הצלעות).

אבחנה: מספר המסלולים הוא 2^n ולכן אי אפשר לעבור על כולם.

פתרון: לכל קודקוד v נגדיר $l(v)$ להיות משקל המסלול המינימלי מ־s ל־v.
יש קשר רקורסיבי בין תתי הבעיות

$$\begin{aligned} l(a_i) &= \min\{l(a_{i-1})+x_{i-1}, l(b_{i-1})+w_{i-1}\} \\ l(b_i) &= \min(l(b_{i-1})+y_{i-1}, l(a_{i-1})+z_{i-1}) \\ l(t) &= \min\{l(a_n)+x_n, l(b_n)+y_n\} \end{aligned}$$

נכונות: באינדוקציה על i:

1. $l(a_1), l(b_1)$ - ברור כי יש מסילה יחידה מ־s ל־a_1 ו־b_1.
2. צ"ל כי $l(a_i)$ זה משקל המסלול המינימלי מ־s ל־a_i. נניח שלא - יהי P מסלול קצר יותר מ־s ל־a_i אז P עובר או ב־a_{i-1} או ב־b_{i-1}.
בה"כ נניח ש־P עובר ב־b_{i-1}, אז ם משרה מסילה P' מ־s ל־b_{i-1}.

$$Length(P') = Length(P) - \omega_{i-1} \le l(a_i) - \omega_{i-1} \le (l(b_{i-1}) + \omega_{i-1}) - \omega_{i-1}$$

סתירה להנחת האינדוקציה.
באופן דומה ניתן להראות ש־$l(b_i)$ אופוטמלי ולכן $l(t)$ אופטימלי.

פתרון נוסף: (פתרון אלטרנטיבי שלי) פתרון פחות או יותר זהה, אך לדעתי יותר מובן ופשוט. לכל קודקוד יש שני אפשרויות התקדמות שנסמנם ב־$L(x_i), R(X_i)$ (באנלוגיה לבן ימני ושמאלי של עץ בינארי), ואז המרחק המינימלי מהקודקוד ה־x_i עד לנקודת הסוף הוא (כאשר $\omega(x)$ הוא המשקל של הצלע המכוונת אל x)

$$l(x_i) = \min\{l(L(x_i)) + \omega(L(x_i)), l(R(x_i)) + \omega(R(x_i))\}$$

הסיבה למשוואה הזו ברורה, אם ידועה לנו המסלול הקצר ביותר עבור שתי האפשרויות לצעד הבא במסלול, אז פשוט ניקח את המסלול הקצר מבינהם.

11.1.1 שיטה - Optimal Substructures

זיהוי אוסף של תתי בעיות כך ש:

1. אפשר לסדר את תתי הבעיות כך שבהינתן הפתרון לבעיות הקודמות, אפשר לחשב את הפתרון לבעייה הבאה.
2. חישבו כל תתי הבעיות מניב פתרון לבעיה המקורית

11.2 תת מחרוזת משותפת ארוכה ביותר (תמא"ב)

הגדרה 11.1 אם $A = (x_1, \ldots, x_n)$ מחרוזת אז $C = (z_1, \ldots, z_k)$ תת מחרוזת של A אם קיימים $1 \le i_1 < \ldots < i_k \le n$ כך ש־$Z_j = X_{ij}$ לכל $1 \le j \le k$.

הערה 11.2 תת המחרוזת לא חייבת להופיע ברצף בתוך A.

הבעיה: בהינתן שתי מחרוזות $A = \{x_1, \ldots, x_n\}, B = \{y_1, \ldots, y_m\}$, מהי תת המחרוזת המשותפת הארוכה ביותר שלהם? כלומר נחפש $Z = \{z_1, \ldots, z_k\}$ כך ש־k מקסימלי.

חלוקה לבעיות: נגדיר ל־$1 \leq i \leq n$ ו־$1 \leq j \leq m$ את $A_i = (x_1, \ldots, x_i)$ ו־$B_j = (y_1, \ldots, y_j)$, ו־$f(i,j)$ הוא אורך התמא"ב של A_i, B_j ולכן סה"כ יש nm בעיות.

הערות:

1. $f(0,j) = f(i,0) = 0$ לכל i, j.
2. $f(n,m)$ היא הבעיה שאנו מנסים לפתור.

בניית המחרוזת: הקשר הרקורסיבי בין תתי הבעיות:

$$f(i,j) = \begin{cases} 1 + f(i-1, j-1) & x_i = y_j \\ \max\left(f(i, j-1), f(i-1, j)\right) & x_i \neq y_j \end{cases}$$

הפתרון: נפתור את תתי הבעיות אחת אחת ונגדיר טבלה בגודל $(n+1)(m+1)$, ובכל תא נכתוב את גודל המחרוזת המקסימלי עבור המקרה ה־i, j ומאיזה מיקום נלקחה האות (על מנת שנוכל לשחזר את הפתרון).

הערה 11.3 אם זוכרים את הבחירות בכל שלב במילוי הטבלה, אז אפשר לשחזר תת המחרוזת הארוכה ביותר.

11.3 בעיית ריבוע הבד

לחנות יש בד בגודל $n \times m$, אבל היא רוצה למכור רק חתיכות בגודלים $(n_1, m_1), \ldots, (n_k, m_k)$, כאשר לכל גודל יש מחיר p_i המתאים לו, ובנוסף מתקיים $n_i < n, m_i < m$.

הנחות:

1. חיתוך חייב להיות דיסקרטי וטוטאלי.
2. המחיר של חתיכת בד שאי אפשר לקבל ממנה את אחד הטיפוסים הוא אפס.
3. אי אפשר לקבל אף אחד מהאיברים המקוריים ע"י חיתוך של גודל מקורי אחר ולהרוויח (אפשר להיפתר מההנחה הזו ולהוסיף עוד איבר למקסימום שיבדוק מה עדיף)

תתי הבעיות: אם $0 \leq x \leq n$ ו־$0 \leq y \leq m$ ו־$P(x,y)$ הוא הערך של מלבן בד בגודל $x \times y$.

בניית הבעיה: הקשר הרקורסיבי בין תתי הבעיות:

$$P(x,y) = \max\left\{\max_{1 \leq k \leq x-1}\left(P(x-k, y) + P(k, y)\right), \max_{1 \leq k \leq y-1}\left(P(x, y-k) + P(x, k)\right), 0\right\}$$

מקרי הבסיס הם:

1. אם $x = 0$ או $y = 0$ אז $P(x,y) = 0$.
2. אם $P(x_i, y_i) = p_i$.

12 תרגול 4 – 02.11.2010

12.1 אלגוריתמים דינמיים

12.1.1 אלגוריתם Floyd Warshall

נתון $G=(V,E)$ גרף מכוון ו־$V=\{v_1,\dots,v_n\}$, וגם פונקציית משקל $\omega : E \to \mathbb{R}$.

המשימה: לכל i,j למצוא את אורך המסילה המינימלית מ־v_i ל־v_j.

הנחה: אין מסילה מעגלית שסכום המשקולות של צלעותיה שלילי.

תתי הבעיות: נגדיר $f(i,j,k)$ להיות אורך המסילה הקצרה ביותר מ־v_i ל־v_j שמשתמשת רק ב־$\{v_1,\dots,v_k\}$ כקודקודי הביניים.
נשים לב כי מספר תתי הבעיות הוא $O(n^3)$, ו־$f(i,j,n)$ היא הבעיה המקורית שאנו רוצים לפתור.

קשר רקורסיבי:

$$\begin{aligned}
f(i,i,k) &= 0\\
f(i,j,0) &= \begin{cases}\omega(v_i,v_j) & (v_i,v_j)\in E\\ \infty & Otherwise\end{cases}\\
f(i,j,k) &= \min\{f(i,j,k-1), f(i,k,k-1)+f(k,j,k-1)\}
\end{aligned}$$

הוכחה: מקרי הבסיס ברורים ונניח נכונות עבור $k-1$ ונראה עבור k.
נתבונן במסילה הקצרה ביותר מ־v_i ל־v_j שמשתמשת רק ב־$v_1,\dots,v_k$ כקודקוד ביניים. מסילה זו שעובר ב־v_k או שלא:

1. אם לא, אז היא המסילה הקצרה ביותר מ־v_i ל־v_j שעוברת רק ב־$v_1,\dots,v_{k-1}$ כלומר $f(i,j,k-1)$.
2. אם כן, אז אורך המסילה שווה לאורך המסילה מ־v_i ל־v_k ועוד אורך המסילה מ־v_k ל־v_j.
וסכום זה יהיה מינימלי אם"ם שניהם המחוברים מינימליים, כלומר נקבל $f(i,k,k)+f(k,j,k)$.
אף אחד מהמחוברים לא יכול להשתמש ב־v_k כקודקוד ביניים, כי אז היה מעגל שלילי ולכן נקבל $f(i,k,k-1)+f(k,j,k-1)$.

■

האלגוריתם: תחילה נחשב את $f(i,j,0)$ לכל i,j. כעת עבור k שרץ מ־1 ל־n נחשב בכל שלב את $f(i,j,k)$ לכל i,j בעזרת הנוסחה.
לבסוף $f(i,j,n)$ זה הפתרון.

זמן ריצה: $O(n^3)=O(1)\cdot O(n^3)$.

12.1.2 בעיית מסילות הרכבת

תיאור הבעיה: נתון מספר שלם $L>0$ שהוא אורך המסילה ו־n סוגי מקטעים: $(d_1,l_1,r_1),\dots,(d_n,l_n,r_n)$, כך ש־$d$ הוא אורך המקטע (טבעי), l סוג מקטע שמאלי ו־r הוא סוג מקטע ימני.

המטרה: למצוא רשימת אינדקסים $s_1, s_2, \ldots, s_k$ כך ש־$d_{s_1} + \ldots + d_{s_k} = L$ ולכל $1 \leq j \leq k-1$ מתקיים כי $r_{s_j} = l_{s_{j+1}}$ (כלומר החיבורים מתאימים), כך ש־k מינימלי (מספר מינימלי של חלקים).

ניסיון ראשון: נגדיר $f(l)$ להיות המחיר (כמות החלקים) המינימלי של מסילה באורך $l \in \mathbb{N}$ ואז $f(l) = \min_{1 \leq j \leq n} (f(l - d_j) + 1)$.
אבל הבנייה הזו לא בהכרח נותנת פתרון תקין כי לא אילצנו את החלק הראשון ששמנו שיתאים לשאר החלקים.

ניסיון שני: נגדיר $f(l,i)$ שהוא המחיר המינימלי של מסילה באורך l המסתיימת בסוג חיבור v_i.

הקשר הריקורסיבי: $f(l,i) = \min_{\substack{1 \leq j \leq n \\ r_j = v_i \\ d_j \leq l}} |f(l - d_j, l_j) + 1|$ כאשר $f(0,i) = 0$ לכל i (וכאשר נגדיר $\min(\emptyset) =$ ∞ אם לא קיים שום פתרון)

מספר תתי הבעיות: $L \cdot m$

זמן ריצה: $O(n)Lm = O(nmL)$. עם ניתוח יותר קפדני, אפשר לקבל $O(L(m+n))$.

הפתרון: ניקח מינימום על כל i עבור $f(L,i)$.

12.2 רשתות זרימה

הגדרה 12.1 רשת זרימה היא גרף מכוון $G = \langle V, E \rangle$ עם 2 קודקודים מיוחדים, s שיקרא קודקוד המקור (Source) ו־t שיקרא קודקוד בור (Sink או Target), ופונקציה $c : E \to \mathbb{R}^+$.

הנחות:

- אם $(u,v) \notin E$ אז $c(u,v) := 0$.
- אין צלעות שנכנסות ל־s ואין צלעות שיוצאות מ־t.

הגדרה 12.2 זרימה היא פונקציה $f : V \times V \to \mathbb{R}$ כך ש־

1. $f(u,v) = -f(v,u)$ לכל u ו־v ב־V.
2. $f(u,v) \leq c(u,v)$ הם אילוצי קיבולת.
3. לכל $s,t \neq v \in V$ מתקיים $\sum_{u \in V} f(u,v) = 0$.

הגדרה 12.3 הגודל של הזרימה (או השטף) מוגדר להיות $|f| = \sum_{u \in V} f(s,u)$.

תכונות: (הוכח בהרצאה)

1. אם $c(u,v) \in \mathbb{N}$ לכל u,v אז יש זרימה f בעלת גודל מקסימלי כך ש־$f(u,v) \in \mathbb{Z}$ לכל $u,v \in V$.
2. עבור חתך של קבוצות זרות A ו־B של קודקודי הגרף נגדיר $c(A,B) = \sum_{u \in A, v \in B} c(u,v)$ (כלומר סכום הצלעות העוברות מצד אחד לצד שני).
3. הזרם בחתך המינימלי שווה לזרימה המקסימלית.
תוצאה: מציאת חתך מינימלי (כלומר חתך בעל זרם מינימלי).
הערה: מציאת חתך מקסימלי היא בעיית NP.

12.2.1 שימוש ברשת זרימה

נסתכל על מטריצה עם n שורות ו־m עמודות.

בעיה: נתונים $r_1,\ldots,r_n \in \mathbb{N}\bigcup\{0\}$ ו־$c_1,\ldots,c_m \in \mathbb{N}\bigcup\{0\}$ ומחפשים מטריצה $n\times m$ של מספרים שלמים אי שליליים כך ש־

1. סכום השורה ה־i שווה ל־v_i.
2. סכום העמוד ה־j שווה ל־c_j.
3. $M=\sum c_j=\sum r_i$.

תרגום הבעיה: נמיר את הבעיה לרשת זרימה, כאשר יש קודקוד לכל שורה R_i ולכל עמודה C_i ועוד קודקוד מקור s וקודקוד בור t.
לכל $c(s,R_i)=r_i$ ו־$c(R_i,C_k)=\infty$ ו־$c(C_j,t)=c_j$.
קל לראות כי החתך המינימלי יהיה M (כי כל חיתוך המכיל את הצלעות בין שורות לעמודות הוא בהכרח אינסופי).
מכך נובע כי יש זרימה בגודל M וכי אפשר בה"כ להניח שכל הזרימה על הצלעות הן מספרים שלמים.
נבנה מטריצה $A=(a_{ij})$ כך ש־ $a_{ij}=f(R_i,C_j)$ ־ מדוע בחירה זו מתאימה? $\sum_{j=1}^{m}a_{ij}=\sum_{j=1}^{m}f(R_i,C_j)=r_i$.

מסקנה: לכל $r_1,\ldots,r_n,c_1,\ldots,c_m$ מספרים טבעיים כך ש־$\sum^n r_i=\sum^m c_j$ יש מטריצה $n\times m$ של מספירם שלמים אי שליליים כך שסכום השורה ה־i שווה ל־r_i וסכום העמודה ה־j היא c_j.

12.2.2 זיווג בגרף דו צדדי

נתון גרף דו צדדי כאשר יש צלעות המחוברות בין צד L ל־R ובנוסף נוסיף קודקוד s המחובר לקודקודים ב־L וקודקוד t המחובר ל־R.

נגדיר $c(s,l_i)=1$ ו־$c(r_j,t)=1$ ועבור $c(l_i,r_j)=1$.

החתך המינימלי הוא M גורר כי הזרימה המקסימלית M ולכן $|f|=M$ שמשתמשת רק במספרים שלמים, ולכן f משתמשת ב־M צלעות אמצעיות ולכן זהו זיווג בגרף.

המשך בתרגול הבא!

13 תרגול 5 – 09.11.2010

13.1 זיווג מקסימלי ־ המשך

בעיה: בהינתן גרף דו צדדי $G=\langle L\bigcup R,E\rangle$, מצא זיווג M ב־G עם מספר צלעות מקסימלי.

פתרון: נגדיר רשת זרימה על הקודקודים $L\bigcup R\bigcup\{s,t\}$.
יש צלעות מ־s לכל קודקוד ב־L ומכל קודקוד ב־R ל־t, ובנוסף יש צלע (u,v) עבור $y\in L,v\in R$ כך ש־ $(u,v)\in E$.
כל הקיבולות ברשת הם 1.

טענה 13.1 אם f זרימה מקסימלית ו־M זיווג מקסימלי, אז נראה ש־$|M|=|f|$.

הוכחה: נוכיח בשלבים.

1. $|M| \le |f|$ ותהי f זרימה מקסימלית.
נניח ש־f זרימה שלמה, ונגדיר את הזיווג M' להיות הצלעות מ־L ל־R ש־f זורם דרכם, וקל לראות ש־$|M'| = |f|$,
ו־M' זיווג חוקי, כי אם לשתי צלעות ב־M' עם קודקוד משותף, אז חוק שימור החומר של f לא מתקיים ־ סתירה!

2. יהי M זיווג מקסימלי, ונגדיר זרימה f שמזרימה 1 מ־s לכל קודקוד ב־L שמזווג ב־M.
נזרים 1 על כל הצלעות ב־M ו־1 מקודקוד ב־R שמזווגים ב־M ל־t.
נסמן זרימה זו ב־f' ולכן $|f| \ge |f'| = |M|$.

3. משתי הטענות נובע כי $|f| = |M|$.

■

זמן הריצה: זמן הריצה הוא $O(|V| \cdot |E|)$ כי למצוא מסלול הוא $O(|E|)$ ומאלגוריתם FF צריך לחפש $O(|V|)$ מסלולים.

13.2 שיטת FF

13.2.1 למה השיטה הנאיבית הפשוטה נכשלת?

אלגוריתם נאיבי: נחפש בכל שלב מסלול מ־s ל־t שניתן להזרים עליו זרימה, ונזרים כמה שאפשר.

דוגמא נגדית: נתונים 4 קודקודים $\{s,t,a,b\}$ והקשתות $\{(s,a,20),(s,t,10),(s,b,10),(b,t,20),(a,b,20)\}$ (כאשר המספר הוא הקיבול של הצלע).
בשלב הראשון נבחר למשל את המסלול $sabt$, אבל אחרי שלב זה לא ניתן לבנות שום מסילה על צלעות לא רוויות.
אם היינו משתמשים ברשת השיורית, אז היה ניתן להמשיך ולתקן את הזרימה ע"י מסלול $sbat$ אשר עובר על הכיוון ההפוך של הצלע (a,b), וכך אנו משפרים את הזרימה מ־20 ל־30.

13.2.2 מה זה רשת שיורית?

הגדרה 13.2 בהינתן זרימה f על G, נגדיר את G_f ע"י $V_F = V$ והצלעות

$$\begin{aligned} E_1 &= \{\overrightarrow{e} | f(\overrightarrow{e}) < c(\overrightarrow{e})\} \\ E_2 &= \{\overleftarrow{e} | f(\overrightarrow{e}) > 0\} \end{aligned}$$

ונגדיר קיבול חדש לצלעות לפי

$$c(e) = \begin{cases} c(e) - f(e) & e \in E_1 \\ f(e) & e \in E_2 \end{cases}$$

13.2.3 מתי FF מקרטע?

דוגמא: אם $E = \{(s,a,1000),(a,t,1000),(s,b,1000),(b,t,1000),(a,b,1)\}$ אז לאלגוריתם FF יכול לבחור מסלולים כך שיקח לו זמן רב להגיע לזרימה האופטימלית, ולכן נרצה אלגוריתם שלא תלוי במשקולות של הצלעות.

אלגוריתם: (Edmonds-Karp) בכל שלב לוקחם את המסליל הקצר ביותר מ־s ל־t ברשת השיורית ואז זמן הריצה הוא $O\left(|V||E|^2\right)$.

13.2.4 איך מוצאים חתך בעל קיבולת מינימלית?

נתונה רשת זרימה G ונתונה זרימה מקסימלית f, ונגדיר $A = \{v \in V | \exists path(s, v)\}$ ו־$B = V \backslash A$ ונראה כי (A, B) הוא החתך המינימלי.

הערות:

1. אין מסלול מ־s ל־t ב־G_f כי אחרת היינו יכולים להגדיל את הזרימה $\Leftarrow$ $t \notin A$.
2. אין צלעות מ־A ל־B ב־G_f.
3. $f(e) = c(e)$ עבור e מ־A ל־B.
4. $f(e) = 0$ עבור כל צלע e מ־B ל־A.

מסקנה 13.3 $|f| = c(A, B)$ ולכן זהו החתך המינימלי (ממשפט Min Cut Max Flow).

13.3 בעיית השחקנים והמשקיעים

נתונים: קבוצת שחקנים $A = \{a_1, \ldots, a_n\}$ ולכל שחקן יש משכורת מתאימה $(s_1, \ldots, s_n)$.
יש קבוצת משקיעים $I = \{I_1, \ldots, I_k\}$ ולכל משקיע יש את הסכום שהוא מוכן להשקיע $(r_1, \ldots, r_k)$.
עבור כל משקיע יש בנוסף קבוצה $F_j \subset A$ המכילה את השחקנים שרק אם כולם ישחקו בסרט, אז המשקיע יהיה מוכן להשקיע בסרט.

נגדיר רווח: $\sum_{j \in I_j \text{ invest}} r_j - \sum_{i | a_i \text{ plays}} s_i$.

המטרה: אנו מחפשים את קבוצת השחקנים והמשקיעים כך שהרווח יהיה מקסימלי.

פתרון: נגדיר רשת זרימה ע"י $V = \{s, t\} \bigcup A \bigcup I$ וקיימת צלע בין s ל־I_i שקיבולה r_i, בין a_i ל־t שקיבולה s_i ובין I_i ל־a_j אם $a_j \in F_i$ וקיבולה אינסוף.

1. איך ניראה החתך המינימלי בגרף זה? (נסמן חתך זה ב־(X, Y))
   (א) אף צלע אמצעית לא שייכת לחתך, כלומר $I_j \in X$ גורר כי $a_i \in X$ עבור כל שחקן $a_i \in F_j$.
   (ב) קבוצת השחקנים ב־X היא האיחוד של ה־F_j של המשקיעים ב־X.
   כי אם a שחקן שלא שייך לאף F_j של משקיע ב־X אז $X \backslash \{a\}$ נותן חתך עם קיבולת $c(X, Y) - s_a$.
2. ולכן כל חתך מינימלי הוא מהצורה

$$X = \{s\} \bigcup (I' \subset I) \bigcup \left(\bigcup_{j | a_j \in I'} F_j \right)$$

3. נעריך את הגודל של חתך כלשהו שיכול להיות מינימלי:

$$\begin{aligned} c(X, Y) &= \sum_{j | I_j \notin X} r_j + \sum_{I_j \in X, a_i \in F_j} s_i \\ &= \sum_{j=1}^{k} r_j - \sum_{j | I_j \in X} r_j + \sum_{I_j \in X, a_i \in F_j} s_i \\ &= \overbrace{\sum_{j=1}^{k} r_j}^{Constant} - \overbrace{\left(\sum_{j | I_j \in X} r_j - \sum_{I_j \in X, a_i \in F_j} s_i \right)}^{Profit} \end{aligned}$$

ולכן $C(X, Y)$ הוא מינימלי כאשר הרווח הוא מקסימלי.

14 תרגול 6 – 16.11.2010

14.1 משפט Hall

הגדרה 14.1 נתון $G=\langle L\bigcup R,E\rangle$ גרף דו צדדי כך ש־$|R|=|L|$. נאמר ש־G מקיים את תנאי Hall אם לכל $A\subset L$ מתקיים $|N(A)|\geq|A|$ כאשר $N(A)$ קבוצת השכנים של הקודקודים ב־A.

משפט 14.2 (משפט Hall) אם G מקיים את תנאי Hall אז קיים ב־G זיווג מושלם.

תזכורת: בשביל לפתור את בעיית הזיווג המושלם באמצעות רשתות זרימה, הגדרנו את הרשת עם אותם הקודקודים של G וגם s,t כאשר בין s ל־L יש קשת בגודל 1, בין L ו־R יש קשת בגודל 1 אס"ם קיימת קשת ב־G ובין R ל־t יש זרימה בגודל 1.

הוכחה: נתאים ל־G רשת זרימה כמו שעשינו בתרגול הקודם.
הראינו בתרגול הקודם כי הזיווג המקסימלי שווה לגודל הזרימה המקסימלית, ולכן מספיק להראות שיש זרימה בגודל n ולכן מספיק להראות שגודל החתך המינימלי הוא n.
ברור כי גודל החתך המינימלי קטן או שווה ל־n ולכן נראה כי הוא גדול או שווה ל־n.
יהי (A,B) חתך כך ש־$t\in A$ ו־$s\in B$. נרצה להראות ש־$n\leq c(A,B)$.
בגלל שכל הקיבולות הם 1, אז מספיק להגדי שמספר הצלעות מ־A ל־$B\geq n$.
יש 3 סוגים של צלעות כאלה, כאשר נסמן $Q=A\bigcap L$ ו־$T=A\bigcap R$.

1. צלעות מ־s ל־$L\backslash Q$ - ומספר הצלעות הוא $|L\backslash Q|=|L|-|Q|=n-|Q|$
2. צלעות מ־Q ל־$R\backslash T$. - ומספר הצלעות $\leq$ מספר השכנים של Q ב־$R\backslash T$ ומספר זה ב־$R\backslash T$ הוא $|N(Q)\backslash T|$ ומתקיים
$$|N(Q)\backslash T|\geq|N(Q)|-|T|\overset{Hall}{\geq}|Q|-|T|$$
3. צלעות מ־T ל־t - ומספר הצלעות הוא $|T|$.

בסה"כ קיבלנו כי $n=(|Q|-|T|)+|T|+|Q|-n$ ולכן יש לכל הפחות n צלעות ב־(A,B) ולכן $n\leq c(A,B)$. ■

14.2 בעיית נקיון אולם ההרצאות

נתונים:

- n סטודנטים $\{s_1,\ldots,s_n\}$.
- m ימים $\{d_1,\ldots,d_m\}$.
- ביום ה־j מגיעים סטודנטים $S_j\subseteq\{s_1,\ldots,s_n\}$.
- הסטודנט ה־i מגיע בימים $D_i\subseteq\{d_1,\ldots,d_m\}$.
- בכל יום מגיע לפחות סטודנט אחד.

מטרה: רוצים לבחור סטודנט $x_j\in S_j$ עבור $j=1,\ldots,n$ שינקה את אולם ההרצאות.
נגדיר $P_i=\sum_{d_j\in D_i}\frac{1}{|S_j|}$ ו־$P_i'=\lceil P_i\rceil$ ונדרוש פתרון בו הסטודנט ה־i מקנה לכל היותר P_i' פעמיים.

פתרון:

1. נגדיר רשת זרימה.

$$V = \{s,t\} \bigcup \{s_1,\ldots,s_n\} \bigcup \{d_1,\ldots,d_m\}$$

נגדיר גם את הקשתות הבאות:

(א) בין s ל־s_i תהיה צלע עם קיבול P'_i.

(ב) בין כל s_i תהיה לצלע ל־ $d_j \in D_i$ בעלת קיבול אינסופי.

(ג) בין d_j ל־t תהיה צלע עם קיבול 1.

2. אם ידוע שגודל הזרימה המקסימלית הוא m, אז היה קיים פתרון, כי לפי למה שהראינו בהרצאות אז קיימת f^* זרימה שלמה בגודל m.
נגדיר שהסטודנט s_i מנקה בימים שאליהם יש ממנו זרימה.
לכל יום נכנס זרימה (כי $m = |f^*|$) מבדיוק סטודנט 1 ולכן זהו פתרון חוקי, ובנוסף הזרימה שנכנסת ל־s_i היא קטנה או שווה ל־P'_i ולכן כל סטודנט מנקה לכל היותר P'_i ימים כמבוקש.

3. גודל הזרימה המקסימלית $\geq m$ כי $m \geq \min cut$ למשל עבור החתך $\{t\}, \{s\} \bigcup L \bigcup R$.

4. נגדיר זרימה f ע"י כל סטודנט המזרים $\frac{1}{|S_j|}$ לכל יום ב־D_i.
נחשב כמה זרימה יוצאת מסטודנט s_i הוא $P'_i \geq P_i = \sum_{d_j \in D_i} \frac{1}{|S_j|}$.
נחשב כמה זרימה נכנסת ליום ה־j הוא $1 = |S_j| \cdot \frac{1}{|S_j|}$.
כל הצלעות (d_j, t) רוויות ולכן הזרימה ברשת האי m.

14.3 אלגוריתמי קירוב

הגדרה 14.3 בעיית אופטימיזציה היא בעיה כך שקיימת קבוצת פתרונות חוקיים P, ופונקציית משקל $f : P \to \mathbb{R}$, כאשר המטרה היא למצוא $p \in P$ כך ש־$f(p)$ מקסימלי או מינימלי (בהתאם לשאלה).

הגדרה 14.4 אלגוריתם c מקרב הוא אלגוריתם שמוצא פתרון $p \in P$ כך שלכל היותר מתקיים $f(p) \geq c \cdot f(p_{opt})$ (עבור בעיית מקסמיזציה $c < 1$) ועבור מינימיזציה $c > 1$.

14.3.1 בעיית כיסוי הקודקודים Vertex Cover

בעיה: נתון גרף $G = \langle V, E \rangle$ ומחפשים ת קבוצה $S \subset V$ קטנה ככל האפשר כך שכל $\{u,v\} \in E$ אז $u \in S$ או $v \in S$.

אלגוריתם: נאתחל $E' = E$, ו־$S = \emptyset$.
בכל שלב נבחר צלע $\{u,v\} \in E'$ ונצרף את u, v ל־S ונמחק מ־E' את כל הצלעות שנוגעות ב־u או ב־v.

נכונות: נניח שהצלעות שהאלגוריתם בוחר הם $e_1 = \{u_1,v_1\}, \ldots, e_n = \{u_n,v_n\}$ אז יש $2n$ קודקודים (כי אחרי בחירת צלע נמחקים כל הצלעות עם הקודקודים הנבחרים).
יהי $U \subset V$ פתרון אופטימלי. U חוקי ולכן מכסה את $e_1, \ldots, e_n$, (שהם כאמור צלעות בלי קודקודים משותפים) ולכן לכל $1 \leq i \leq n$ אז $u_i \in U$ או $v_i \in U$, ולכן $|U| \geq n$.

14.3.2 בעיית מיקום הקניונים (מתוך הספר של Tardos פרק 11)

יש n ערים $S = \{s_1, \ldots, s_n\}$ במישור ואנו רוצים לבנות k קניונים, בצורה כזאת שהמרחק המקסימלי מעיר לקניון יהיה מינימלי (נניח $k \leq n$).

פתרון הוא סדרה של k מיקומים $C=\{c_1,\ldots,c_k\}$ וערך של פתרון הוא

$$\begin{aligned} r(C) &= \max_{1\le i\le n}\left(d\left(s_i,C\right)\right) \\ d\left(s_i,C\right) &= \min_{1\le j\le k}\left(d\left(s_i,c_j\right)\right) \end{aligned}$$

נניח שהיינו יודעים ש־$r\left(C_{opt}\right)=\mu$. נאתחל $S'=S$ ו־$C=\emptyset$.
בכל שלב נבחר $s_i\in S'$ ונצרף אותה ל־C. נסלק מ־S' את הערים שמרחקם הוא $2\mu\ge s_i$.
ברור ש־C הוא 2 קירוב, אבל לא ברור ש־C הוא פתרון חוקי כי אולי $|C|>k$.
נניח בשלילה ש־$|C|>k$. נניח ש־$C=\{c_1,\ldots,c_m\}$.

למה: לכל i,j מתקיים כי $2\mu<d\left(c_i,c_j\right)$.

הוכחה: נניח $i<j$ ולכן בחרנו את c_i קודם ואז סילקנו את כל המקומות במרחק $2\mu \ge$.

הערה: לכל c_i יש יש c_j^* במרחק $\mu \ge$ ממנו, ולכן אין c_j^* שקרוב ל־2 $c_i,c_j\in C$ (מא"ש המשולש).

ולכן $k<m\le\left|C_{Opt}\right|$ סתירה!

האלגורתם השלם: (מבלי לדעת את μ)

1. נאתחל את $C=\emptyset$.
2. בוחרים $s\in S$ כלשהו ומצרפים אותו ל־C.
3. בכל שלב מוציאים $s\in S$ כך ש־$d\left(s,C\right)$ מקסימלי ומצרפים אותו ל־C.

טענה 14.5 זהו אלגוריתם 2־קירוב.

הוכחה: נראה שהאלגוריתם הוא ריצה חוקית של האלגוריתם הקודם באינדוקציה.
בשלב ביניים באלגוירתם, נניח שקיים $s\in S$ במרחק $2\mu <$ מ־C, אז גם האלגוריתם החדש יבחר קודקוד במרחק $2\mu <$.
■

15 תרגול 7 – 23.11.2010 – אלגוריתמי קירוב רנדומיים

15.1 תזכורת מושגים בהסתברות

מבצעים ניסוי E שהוא אקראי (כמו הטלת קוביה או מטבע), ושיש לו מרחב מדגם $\Omega=\{\omega_1,\ldots,\omega_n\}$.

הגדרה 15.1 משתנה מקרי X הוא פונקציה $X:\Omega\to\mathbb{R}$.

הגדרה 15.2 תוחלת של מ"מ X שמקבל ערכים $x_1,\ldots,x_n$ בהסתברויות $p_1,\ldots,p_n$ בהתאמה, מוגדרת ע"י

$$E\left(x\right)=\sum_{i=1}^{n}p_ix_i$$

הערה 15.3 התוחלת של X היא בהסתברות גבוהה קירוב טוב לממוצע של X על גבי ניסויים רבים.

לינאריות התוחלת: אם $X=Y+Z$ אז

$$E(X)=E(Y)+E(Z)$$

הכללה: אם $X=\sum_{i=1}^{n}X_i$ אז $E(X)=\sum_{i=1}^{n}E(X_i)$.

א"ש מרקוב: יהי X משתנה מקרי אי שלילי. $0\geq X$ עם תוחלת $E(X)=\mu$, אזי עבור $c>1$

$$P(X\geq c\cdot\mu)\leq\frac{1}{c}$$

דוגמא: מטילים מטבע לא הוגן שהוא 1 בהסתברות $\frac{1}{100}$ ו־0 בהסתברות $\frac{99}{100}$ אזי

$$E(X)=1\cdot\frac{1}{100}+0\cdot\frac{99}{100}=\frac{1}{100}$$

ומא"ש מרקוב נובע כי

$$P(X\geq 1)=P(X\geq 100\mu)\leq\frac{1}{100}$$

הגדרה 15.4 נאמר שמ"מ X,Y הם ב"ת אם לכל $x,y\in\mathbb{R}$ אז

$$P(X=x,\wedge Y=y)=P(X=x)P(Y=y)$$

אינטואטיבית: ל־X אין שום קורולציה עם Y, או X לא תורם מידע נוסף על Y.

הערה 15.5 נניח שיש ניסוי שמצליח בהסתברות $\epsilon>0$ ונכשל בהתסברות $(1-\epsilon)$.
נניח שריצות נפרדות של הניסוי הן ב"ת, אז ההסתברות שאחרי k ריצות כולן נכשלו היא $(1-\epsilon)^k$.
נסמן $k=\frac{t}{\epsilon}$ ואז

$$\begin{aligned}(1-\epsilon)^k &= \left((1-\epsilon)^{\frac{1}{\epsilon}}\right)^t\\ &\leq e^{-t}\end{aligned}$$

מסקנה 15.6 אם רוצים שניסוי יצליח בסיכוי $\leq 1-e^{-t}$ אז מספיק להריץ את הניסוי $k=\left\lceil\frac{t}{\epsilon}\right\rceil$ פעמיים.

מודל 1: אלגוריתם הסתברותי ־ בהינתן קלט תחזיר תשבוה נכונה בהסתברות $\leq\epsilon+\frac{1}{2}$.

הערה 15.7 קבוצת הבעיות שהמודל הנ"ל פותר נקראות BPP.

מודל 2: אלגוריתם רנדומי מקבל קלט, אם התשובה היא לא, הוא יחזיר לא.
אם התשובה היא כן, אז הוא יחזיר כן בסיכוי $\leq\epsilon$ עבור איזשהו ϵ.

הערה 15.8 המחלקה של בעיות שהמודל הנ"ל פותר נקראות RP.

הגדרה 15.9 אלגוריתם C קירוב הסתברותי, הוא אלגוריתם רנדומי שמשיג על כל קלט C קירוב בהסתברות $0<\epsilon$.

15.2 בעיית 3SAT מקסימלי

הגדרה 15.10 משתנה בוליאני הוא משתנה היכול לקבל רק ערך 0 או 1 (אמת או שקר).

אם $x_1, \ldots, x_n$ משתנים בוליאניים, אז ניתן לכתוב בעזרתם ביטויים עם פעולות "או", "וגם" או "שלילה" למשל: $\neg\left(x_1 \wedge \left(x_2 \vee x_3\right)\right)$.

הבעיה: בהינתן ביטוי המורכב ממשתנים ואופרטורים בוליאניים, נרצה לבדוק האם קיים איזשהו צירוף של הצבות ל־$x_1, \ldots, x_n$ כך שהביטוי יהיה חיובי.
בעייה זו נקראת Satisfiability Problem ובקיצור SAT.

עובדה: בהינתן נוסחא בוליאנית ב־n משתנים, הבעיה של "אם קיימת השמה מספקת" היא בעיית NP שלמה.

בהינתן $x_1, \ldots, x_n$ וביטוי מהצור ה

$$\left(\overbrace{x_1 \vee x_3 \vee \ldots \vee \neg x_{17}}^{k}\right) \wedge \left(\overbrace{x_1 \vee \ldots \vee x_n}^{k}\right) \wedge \ldots$$

כלומר בכל ביטוי בסוגריים יש רק אופרטורים של "או" או "שלילה" ובין הסוגריים אופרטור של "גם", כאשר בכל תת ביטוי יש k משתנים.
לצורה זו קוראים $k - CNF$ (הצורה הקנונית).

בעיה מפושטת: $(k - SAT)$ בהינת פסוקית ב־$k - CNF$, האם יש לב השמה מספקת?

הערה 15.11 עבור $k = 1$ זו נחשבת לבעיה קלה.
עבור $k = 2$ קיים אלגוריתם פולינומיאלי.
עבור $k = 4$ ומעלה זוהי בעיית NP שלמה.

דוגמא: למשל בהינתן הביטוי $\left(x_1 \vee \neg x_2 \vee x_3\right) \wedge \left(\neg x_2 \vee \neg x_1 \vee x_3\right)$ אז עבור $x_3 = 1$ נקבל פסוק אמת.

הגדרה 15.12 בעיית Max-k-SAT ־ בהינתן נוסחה ב־$k - CNF$, מצא השמה למשתנים כך שמספר מקסימלי של פסוקיות הן מסופקות.
הערך של הפיתרון הוא מספר הפסוקיות שסופקו.

אלגוריתם הסתברותי: לכל משתנה מטילים מטבע, וקובעים שהוא אפס בהסתברות $\frac{1}{2}$ ו־1 בהסתברות $\frac{1}{2}$.
עבור נוסחה עם m פסוקיות במשתנים $x_1, \ldots, x_n$ ב־$3CNF$.

חישוב התוחלת: נסמן ב־X את מספר הפסוקיות המסופקות, ונרצה לחשב את $E\left(X\right)$.
נגדיר לכל $1 \leq j \leq m$

$$X_j = \begin{cases} 1 & \text{The j-th close is satisfied} \\ 0 & \text{Otherwise} \end{cases}$$

ומתקיים כי $X = \sum_{j=1}^{m} X_j$ ולכן

$$\begin{aligned} E\left(X_j\right) &= 0 \cdot P\left(X_j = 0\right) + 1 \cdot P\left(X_j = 1\right) \\ &= P\left(X_j = 1\right) = \frac{7}{8} \end{aligned}$$

(כאשר מניחים כי בכל פסוקית יש 3 משתנים שונים ־ הנחה סבירה שאפשר לבדוק ב־$O\left(n\right)$)
ולכן בסה"כ נקבל כי

$$E\left(X\right)=\sum_{j=1}^{n}E\left(X_j\right)=\frac{7}{8}m$$

באופן כללי עבור $k-SAT$ נקבל כי $E\left(X\right)=m\left(1-\frac{1}{2^k}\right)$.
אם $m<2^k$ אז מתקיים כי $\left(1-\frac{1}{2^k}\right)m>m-1$.

ההסתברות לקירוב טוב: נרצה לדון בסיכויי ההצלחה, כלומר $P\left(X\geq\frac{7}{8}m\right)$.
נגדיר משתנה מקרי Y להיות מספר הפסוקיות הלא מסופקות, וכיוון ש־$X+Y=m$ אז $E\left(Y\right)=m-E\left(X\right)=\frac{1}{8}m$.
האלגוריתם לא מצליח אם"ם $Y>\frac{1}{8}m$, אז מא"ש מרקוב נקבל

$$P\left(Y>\frac{1}{8}m\left(1+\epsilon\right)\right)\leq\frac{1}{1+\epsilon}$$

אך תוצאה זו לא עוזרת לנו. נשים לב ש־m ו־Y הם מספרים שלמים, ולכן אם $Y>\frac{1}{8}m$ אז $Y\geq\frac{1}{8}m+\frac{1}{8}$ ולכן

$$\begin{aligned}P\left(Y>\frac{1}{8}m\right) &= P\left(Y\geq\frac{1}{8}m+\frac{1}{8}\right)\\ &= P\left(Y>\frac{1}{8}m\left(1+\frac{1}{m}\right)\right)\\ &\overset{Markov}{\leq} \frac{1}{1+\frac{1}{m}}\\ &= \frac{m}{m+1}=1-\frac{1}{m+1}\end{aligned}$$

ולכן האלגוריתם מצליח בסיכוי $\frac{1}{1+m}$.

אלגוריתם משופר: נריץ את האלגוריתם הקודם $m+1$ פעמים, ונבחר את ההשמה שנתנה את הפתרון הכי הטוב.
הסיכוי של אלגוריתם זה להיכשל הוא

$$\left(1-\frac{1}{m+1}\right)^{m+1}\leq\frac{1}{e}$$

ולכן הסיכוי להצליח שואף ל־$\frac{2}{3}\approx1-\frac{1}{e}$.

זמן ריצה: $O\left(\left(m+n\right)m\right)$.

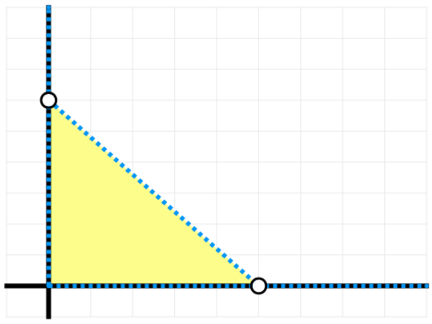
15.3 בעיית Vertex Cover ממושקל בעזרת תכנון לינארי

הבעיה: נתון גרף $G=\left\langle V,E\right\rangle$ לא מכוון עם משקולות $\omega:V\to\mathbb{R}$ (המשקולות על הקודקודים!).
מחפשים כיסוי בקודקודים במשקל מינימלי (בתרגול הקודם פתרנו את הבעיה כאשר $\omega\left(v\right)=1$ לכל v)

תכנון לינארי: יהיו $x_1,\ldots,x_n\in\mathbb{R}$. תהי f פונקציה לינארית ב־$x_1,\ldots,x_n$ משתנים, כלומר $f=\sum_{i=1}^{n}a_ix_i$.
בנוסף נתונים k אילוצים לינאריים על x_i, כלומר ביטוי מהצורה $\sum b_ix_i\leq C$ עבור איזשהו $C,b_i\in\mathbb{R}$ קבועים.

המטרה: מצא את הנקודה $x_1^*,\ldots,x_n^*$ הממקסמת את f בכפוף לאילוצים.

דוגמא: $f=2x+y$ כאשר $x+y\leq 1$, $x\geq 0$ ו־$y\geq 0$.



איור 15.1: תיאור גרפי לבעיה

כאשר מהציור נקבל כי המקסימום הוא אחת הנקודות מתוך השטח הצבוע, ובמקרה זה נקודה בודדת $x=1,y=0$.

עובדה: קיים אלגוריתם פולינומי במספר המשתנים ומספר האילוצים שפותר את בעיית התכנון הלינארי.

ניסוח בתכנון לינארי ל־VC: המשתנים $x_1,\ldots,x_n$ מקיימים $x_i=1$ אם"ם בחרנו את v_i ל־U.
אנו רוצים למקסם את $f\left(x_1,\ldots,x_n\right)=\sum_{i=1}^{n}\omega\left(v_i\right)x_i$.
האילוצים הם $x_i\in\{0,1\}$ ולכל צלע $\{v_i,v_j\}\in E$ מתקיים כי $x_i+x_j\geq 1$.

בעיה יותר כללית: במקום $x_i\in\{0,1\}$ נכתוב $x_i\in[0,1]$.

הערה 15.13 כל פתרון לבעיה הראשונה הוא גם פתרון חוקי (כלומר עומד באילוצים) לבעיה השנייה.

מסקנה 15.14 אם $x_1^*,\ldots,x_n^*$ הוא פתרון אופטימלי לבעיה השנייה, אז $|OPT|\geq f\left(x_1^*,\ldots,x_n^*\right)$ (קטן שווה מהערך של פתרון אופטימלי לבעיה הראשונה).

נראה כי הפתרון שאנו מוצאים הוא קרוב לפתרון הרצוי.
$x_1^*,\ldots,x_n^*$ הם בין 0 ל־1, אם הם כולם ב־$\{0,1\}$ אז סיימנו.
אחרת נרצה "לעגל" אותם. נגדיר

$$x_i'=\begin{cases}1 & x_i^*\geq\frac{1}{2}\\ 0 & Otherwise\end{cases}$$

נראה כי הפתרון החדש הוא חוקי. תהי $(v_i,v_j)\in E$, אז $x_i^*+x_j^*\geq 1$ ולכן לפחות אחד מהם גדול או שווה ל־$\frac{1}{2}$.
נניח בה"כ ש־x_j^*, אז $x_j'=1$ ו־$\{v_i,v_j\}$ מכוסה ־ הפתרון חוקי!
נראה שהפתרון אופטימלי. נניח כעת כי המשקולות הן אי שליליות.

$$\begin{aligned}
|OPT| &\geq f\left(x_1^*,\ldots,x_n^*\right)=\sum_{i=1}^{n}\omega\left(v_i\right)x_i^*\\
&= \frac{1}{2}\sum\omega\left(v_i\right)\left(2x_i^*\right)\\
\omega\left(v_i\right)\geq 0\Rightarrow\quad &\geq \frac{1}{2}\sum\omega\left(v_i\right)x_i'\\
&= \frac{1}{2}|ALG|
\end{aligned}$$

ולכן האלגוריתם הזה הוא 2 קירוב.

16 תרגול 8 – 30.11.2010

16.1 תזכורת על מספרים מרוכבים

הגדרה 16.1 השדה המרוכב הוא $\mathbb{C} = \{a+bi | a,b \in \mathbb{R}\}$ כאשר $i = \sqrt{-1}$ (הראינו בלינארית 1 הגדרה יותר מדוייקת).

ניתן לחשוב על מספר מרוכב כעל וקטור $\begin{pmatrix} a \\ b \end{pmatrix} \iff a+bi$, וכעת עם נעבור לקואורדינטות קוטביות, אז נוכל להביע כל מספר מרוכב ע"י $r(\cos\theta + i\sin\theta) \equiv r \cdot cis(\theta)$.

בהצגה קוטביות קל יותר לכפול מספרים מרוכבים, וניתן להראות שמתקיים

$$r_1 \cdot cis(\theta_1)\, r_2 \cdot cis(\theta_2) = r_1 r_2 \cdot cis(\theta_1 + \theta_2)$$

משפט 16.2 (נוסחת אויילר) ניתן להביע מספר מרוכב $r \cdot cis(\theta)$ ע"י $r \cdot e^{i\theta}$.

הוכחה: באמצעות טורי טיילור נקבל

$$\begin{aligned}
\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \\
\cos x &= 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \dots \\
e^x &= 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \dots \\
\cos x + i\sin x &= ix - i\frac{x^3}{3!} + i\frac{x^5}{5!} - \dots + 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \dots \\
&= ix + \frac{(ix)^3}{3!} + \frac{(ix)^5}{5!} - \dots + 1 + \frac{(ix)^2}{2} + \frac{(ix)^4}{4!} - \dots \\
&= 1 + \frac{(ix)^2}{2} + \frac{(ix)^3}{3!} + \dots \\
&= e^{ix}
\end{aligned}$$

■

עובדה: לכל פולינום (ממעלה > 0) מעל $\mathbb{C}$ יש שורש.

עובדה: לפולינום (לאו דווקא מרוכב) ממעלה $\geq n$ יש לכל היותר n שורשים.

16.2 שורשי היחידה

הגדרה 16.3 שורשי היחידה מסדר n הם הפתרונות של $x^n = 1$, כלומר $1, e^{\frac{1}{n}2\pi i}, e^{\frac{2}{n}2\pi i}, \dots, e^{\frac{n-1}{n}2\pi i}$.

סימון: נסמן $\omega_n = e^{\frac{1}{n}2\pi i}$ ואז $\omega_n^k = e^{\frac{k}{n}2\pi i}$ הוא השורש היחידה מסדר n ה־k.

הגדרה 16.4 שורש יחידה פרמיטיבי מסדר n זה שורש יחידה ω מסדר n, כך ש־$\omega^k \neq 1$ עבור $0 < k < n$.

הגדרה 16.5 (הגדרה שקולה לאחרונה) ω_n^k פרימיטיבי אם"ם ל־n,k אין מחלקים משותפים.

הגדרה 16.6 חבורה היא קבוצה X עם פעולה אותה נסמן ע"י "$\cdot$" כך שמתקיים

1. קיים איבר יחידה $e \in X$ כך שעבור כל $x \in X$ מתקיים כי $x \cdot e = e \cdot x = x$.
2. לכל $x \in X$ קיים $x \in X$ כך ש־$x \cdot y = y \cdot x = e$ ונסמן $y =: x^{-1}$.
3. עבור כל $x, y, z \in X$ מתקיים

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

הערה 16.7 לשים לב שלא נדרשת קומוטטיביות.

טענה 16.8 שורשי היחידה מסדר n עם פעולת כפל מהווים חבורה.

הוכחה: נוכיח שהתכונות מתקיימות:

1. המספר 1 הוא איבר היחידה באופן טריוויאלי.
2. יהי ω_n^k שורש יחידה מסדר n, אזי מתקיים $\omega_n^{n-k} = \left(\omega_n^k\right)^{-1}$ כי

$$\omega_n^k \omega_n^{n-k} = \omega_n^{k+n-k} = (\omega_n)^n = 1$$

3. כפל מעל המרוכבים הוא אסוציאטיבי.

■

למה 16.9 (למת הביטול) מתקיים כי $\omega_n^k = \omega_{dn}^{dk}$.

הוכחה: מחוקי חזקה נובע כי $\omega_n^k = e^{\frac{k}{n}2\pi i} = e^{\frac{dk}{dn}2\pi i} = \omega_{dn}^{dk}$. ■

מסקנה 16.10 מספר מסקנות:

1. n זוגי, אז מתקיים $\omega_n^{\frac{n}{2}} = \omega_2^1 = -1$.
2. טענה: אם מעלים את שורשי היחידה מסדר n (n זוגי) בריבוע, אז מתקבלים שורשי היחידה מסדר $\frac{n}{2}$.
3. סכום שורשי היחידה שווה לאפס: $\sum_{k=0}^{n-1} \omega_n^k = 0$ (עבור $n > 1$).
4. באופן כללי,

$$\sum_{j=0}^{n-1} \left(\omega_n^k\right)^j = \begin{cases} n & k \equiv_n 0 \\ 0 & Otherwise \end{cases}$$

הוכחה: הוכחת המסקנות:

1. ברור.
2. $\left(\omega_n^k\right)^2 = \omega_n^{2k} = \omega_{\frac{n}{2}}^k$.
3. נשים לב שזהו סכום של סדרה הנדסית ולכן מתקיים

$$\sum_{k=0}^{n-1} \omega_n^k = \frac{\omega_n^n - 1}{\omega_n - 1} = \frac{1-1}{\omega_n - 1} = 0$$

4. כמו מקודם

$$k \not\equiv_n 0 \quad \Rightarrow \quad \sum_{j=0}^{n-1} \left(\omega_n^k\right)^j = \frac{\omega_n^{kn} - 1}{\omega_n^k - 1} = \frac{1-1}{\omega_n^k - 1} = 0$$

$$k \equiv_n 0 \quad \Rightarrow \quad \sum_{j=0}^{n-1} \left(\omega_n^k\right)^j = \sum_{j=0}^{n-1} (1)^j = n$$

■

16.3 פולינומים

אם נסתכל על פולינום כללי $p(x) = a_0 + a_1 x + \ldots + a_{n-1} x^n$ אז נוכל לייצג אותו ע"י וקטור מקדמים $(a_0, \ldots, a_{n-1}) = v_p$.

נגדיר את הוקטורים $u_j \in \mathbb{C}^n$ עבור $0 \le j \le n-1$ ע"י $u_j^{(k)} = \omega_n^{k \circ j}$ (הקואורדינטה ה־k של u_j)

תזכורת: מכפלה פנימית מעל המרוכבים מוגדרת ע"י $\langle u, v \rangle = \sum_{i=0}^{n-1} u_i \overline{v_i}$.

נשים לב ש־$\overline{\omega_n^k} = \omega_n^{-k}$ ולכן מתקיים כי

$$\langle v_p, u_j \rangle = \sum_{k=0}^{n-1} a_k \omega_n^{-j \cdot k} = \sum a_k \left(\omega_n^{-j}\right)^k = p\left(\omega_n^{-j}\right)$$

הגדרה 16.11 דרך נוספת להגדיר פולינום היא ע"י וקטור של הצבות ערכים $(p(x_0), \ldots, p(x_{n-1}))$ עבור $x_0, \ldots, x_{n-1} \in \mathbb{C}$ שונים זה מזה.

טענה 16.12 עבור כל וקטור $(y_0, \ldots, y_{n-1}) \in \mathbb{C}^n$ קיים פולינום יחיד p ממעלה $\le n-1$ כך ש־ $p(x_0) = y_0, \ldots, p(x_{n-1}) = y_n$.

הוכחה: נוכל לכתוב

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_{n-1} x_0^{n-1} &= y_0 \\ &\vdots \\ a_0 + a_1 x_{n-1} + \ldots + a_{n-1} x_{n-1}^{n-1} &= y_{n-1} \end{aligned}$$

ונוכל לייצג את מערכת המשוואות הנ"ל ע"י מטריצה

$$\begin{pmatrix} 1 & x_0 & \ldots & x_0^{n-1} \\ \vdots & \ddots & & \vdots \\ 1 & x_{n-1} & \ldots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

וכזכור מלינארית 1, זאת מטריצה ון דר מונדה והדטרמיננטה שלה שונה מאפס, ולכן זאת מטריצה הפיכה, ולכן למערכת המשוואות הזאת קיים פיתרון יחיד. ■

16.3.1 זמן ריצה של פעולות על פולינומים

ייצוג ע"י מקדמים:

1. חיבור: $(a_0, \ldots, a_{n-1}) + (b_0, \ldots, b_{n-1}) = (a_0 + b_0, \ldots, a_{n-1} + b_{n-1})$ ולכן זמן הריצה הוא $O(n)$.
2. כפל: אם מכפילים $(a_0, \ldots, a_{n-1})$ ב־$(b_0, \ldots, b_{n-1})$ אז נרצה למצוא את $(c_0, \ldots, c_{n+m-2})$ (תוצאת ההכפלה) כך ש־

$$c_k = \sum_{j=0}^{k} a_j b_{k-j}$$

ולכן זמן הריצה הוא $O\left((m+n)^2\right)$.

3. הצבה: בהינתן x^* בשביל למצוא את $p(x^*)$ נדרש $O(n)$ פעולות.

ייצוג ע"י ערכים:

1. חיבור: אם נתונים $(p(x_0), \ldots, p(x_{n-1}))$ ו־$(q(x_0), \ldots, q(x_{n-1}))$, אז חיבור הפולינומים הוא פשוט חיבור איבר איבר, ולכן $O(n)$.
2. כפל: אם נתונים פולינומים כמו מקודם, אז כפל פולינומים הוא פשוט כפל איבר איבר.
נניח שהיה ייצוג יתר של p ו־q עם לפחות $m+n$ נקודות, אז אפשר לבצע כפל ב־$O(m+n)$.
3. הצבה: נדרש להעביר את הייצוג לייצוג של מקדמים בשביל לחשב את ההצבה באופן פשוט.

16.3.2 הכפלת פולינומים באופן כללי

בהינתן 2 פולינומים p, q ממעלה $n \geq$ אותם נרצה להכפיל:

1. נציב אותם בייצוג ערכים עם $2n$ ערכים.
2. נכפיל את ייצוג הערכים.
3. נעביר את הייצוג של המכפלה חזרה לייצוג מקדמים.

הערה 16.13 בהרצאות נראה כיצד עוברים בין הייצוגים ב־$O\left(n\log n\right)$.

פתרון נאיבי להעברת ייצוגים:

1. ממקדמים לערכים נדרש $O\left(n\right)$ הצבות בפולינום ממעלה n ולכן $O\left(n^2\right)$.
2. ממערכים למקדמים אפשר ב־$O\left(n^3\right)$ באמצעות הפיכת מטריצה.

16.3.3 קונבולוציה

הגדרה 16.14 יהיו $v=\left(a_0,\ldots,a_{n-1}\right)$ ו־$u=\left(b_0,\ldots,b_{m-1}\right)$ שני וקטורים ונסמן את הקונבולוציה של u ו־v ע"י

$$\begin{aligned} u*v &= \left(c_0,\ldots,c_{m+n-1}\right)\\ c_k &= \sum_{j=0}^{k}a_jb_{k-j}\end{aligned}$$

הערה 16.15 אם u ו־v הם וקטורי מקדמים של שני פולינומים, אז $u*v$ זה וקטור המקדמים של מכפלתם.

שימוש: אם $A,B\subseteq\{1,\ldots,n\}$ ו־$A+B=\{a+b:a\in A,b\in B\}$.
נגדיר וקטורים $\{0,1\}^n$ I_A,I_B ע"י $I_{A_j}=\begin{cases}1 & j<A\\ 0 & Otherwise\end{cases}$.
למשל $A=\{1,3\}$ ו־$B=\{1,3\}$ ו־$I_A=(1,0,1,0,0)$ ו־$I_B=(0,1,1,0,0)$.
I_A*I_B הוא וקטור באורך $2n$ ו־$C_k=\sum_{j=0}^{k}I_{A_j}I_{B_{k-j}}$.
האיברים ששונים מאפס בווקטור I_A*I_B הם איברי $A+B$.
בתנאי שיודעים להכפיל פולינומים ב־$O\left(n\log n\right)$ אז ניתן למצוא את גם חיבור הקבוצה ב־$O\left(n\log n\right)$.

17 תרגול 9 – 07.12.2010

17.1 תזכורת ־ אלגוריתם FFT

בהינתן פולינום p ממעלה $n>$ בעל המקדמים $\left(a_0,\ldots,a_{n-1}\right)$ נרצה לחשב את הוקטור $\left(p\left(\omega_n^0\right),\ldots,p\left(\omega_n^{n-1}\right)\right)$.
נגדיר שני פולינומים

$$\begin{aligned} p_0 &\leftrightarrow \left(a_0,a_2,\ldots\right)\\ p_1 &\leftrightarrow \left(a_1,a_3,\ldots\right)\end{aligned}$$

ואז נוכל לכתוב כי

$$(*)\quad p\left(x\right)=p_0\left(x^2\right)+x\cdot p\left(x^2\right)$$

מכיוון ש־

$$\begin{aligned} p_0\left(x^2\right) &= a_0+a_2x^2+\ldots\\ x\cdot p_1\left(x^2\right) &= a_1x+a_3x^3+\ldots\end{aligned}$$

עתה נרצה להעריך את p בנקודות $\omega_n^0,\ldots,\omega_n^{n-1}$, ולפי $(*)$ מספיק להעריך את p_0,p_1 בנקודות $\left(\omega_n^0\right)^2,\ldots,\left(\omega_n^{n-1}\right)^2$.
אם n זוגי, אז $\left(\omega_n^k\right)^2=\omega_n^{2k}=\omega_{\frac{n}{2}}^k$ ולכן מספיק להעריך את p_0,p_1 ב־$\left\{\omega_{\frac{n}{2}}^0,\ldots,\omega_{\frac{n}{2}}^{\frac{n-1}{2}}\right\}$.

סיכום: בהינתן $(a_0, a_1, \ldots, a_{n-1})$ כך ש-$n = 2^s$ עבור $s \in \mathbb{N}$, נרצה לחשב את $p\left(\omega_n^0\right), \ldots, p\left(\omega_n^{n-1}\right)$.

1. בסיס: אם $n = 1$ החזר את a_0.
2. נגדיר $p_1 \leftrightarrow (a_0, a_2, \ldots, a_{n-2})$ ו־$p_1 \leftrightarrow (a_1, a_3, \ldots, a_{n-1})$.
3. נעריך את p_0 ו־p_1 באופן רקורסיבי ב־$\frac{n}{2}$ שורשי היחידה מסדר $\frac{n}{2}$.
4. עבור $0 \le j \le n-1$ נעריך

$$p\left(\omega_n^j\right) = p_0\left(\omega_{\frac{n}{2}}^j\right) + \omega_n^j p_1\left(\omega_{\frac{n}{2}}^j\right)$$

17.2 דוגמא להרצת FFT

הפולינום: $2x^2 + x - 1$

נעבור לווקטור המקדמים נוסיף חזקה ריקה ע"מ שמספר המקדמים יהיה חזקה של 2, ולכן $p = (-1, 1, 2, 0)$.

$$\begin{aligned} p_0 &\leftrightarrow (-1, 2) \\ p_1 &\leftrightarrow (1, 0) \end{aligned}$$

עבור p_0 נקבל ש־ $p_{00} \leftrightarrow (-1)$ ו־$p_{01} \leftrightarrow (2)$ ועתה נעריך עבור השורשים $1, -1$ ונקבל

$$\begin{aligned} p_0(1) &= p_{00}(1) + 1 \cdot p_{01}(1) = 1 \\ p_0(-1) &= (-1) + (-1) \cdot 2 = 3 \end{aligned}$$

באותו אופן $p_{10} \leftrightarrow (1)$ ו־$p_{11} \leftrightarrow (0)$ ולכן

$$\begin{aligned} p_1(1) &= 1 \\ p_1(-1) &= 1 \end{aligned}$$

עתה נחבר את התוצאות עבור p ונקבל

$$\begin{aligned} p(1) &= p_0(1) + 1 \cdot p_1(1) = 2 \\ p(i) &= p_0(-1) + i \cdot p_1(-1) = -3 + i \\ p(-1) &= p_0(1) + -p_1(1) = 0 \\ p(-i) &= p_0(-1) - i \cdot p_1(-1) = -3 + i \end{aligned}$$

ולכן קיבלנו את וקטור הערכים $(2, -3+i, 0, -3-i)$.

17.3 אלגוריתם הפוך ל־FFT

נשים לב שפעולת ה־DFT זה פשוט כפל במטריצה:

$$\underbrace{\begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & \omega_n^1 & \ldots & \omega_n^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \ldots & \left(\omega_n^{n-1}\right)^{n-1} \end{pmatrix}}_{V} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} p\left(\omega_n^0\right) \\ p\left(\omega_n^1\right) \\ \vdots \\ p\left(\omega_n^{n-1}\right) \end{pmatrix}$$

ולכן נוכל לבצע את הפעולה ההופכה ע"י V^{-1} ע"י

$$V^{-1}\begin{pmatrix} p\left(\omega_n^o\right) \\ \vdots \\ p\left(\omega_n^{n-1}\right)\end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}$$

וניתן להראות שמתקיים ש־

$$V^{-1} = \frac{1}{n}\begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & \omega_n^{-1} & \ldots & \omega_n^{-(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{-(n-1)} & \ldots & \left(\omega_n^{-(n-1)}\right)^{n-1}\end{pmatrix}$$

ולכן כדי לבצע FFT^{-1} מספיק להראות כיצד מכפילים את V^{-1} בוקטור q המייצג פולינום שמקדמיו הם איברי הוקטור.

$$V^{-1}\begin{pmatrix} b_0 \\ \vdots \\ b_{n-1}\end{pmatrix} = \frac{1}{n}\begin{pmatrix} b_0+b_1+\ldots+b_{n-1} \\ \vdots \\ b_0+b_1\omega_n^{n-1}+\ldots+b_{n-1}\left(\omega_n^{-(n-1)}\right)^{n-1}\end{pmatrix} = \begin{pmatrix} q\left(1\right) \\ \vdots \\ q\left(\omega_n^{-(n-1)}\right)\end{pmatrix}$$

דוגמא: נמיר את $2x-3$ לוקטור ערכים ע"י הצבה $\begin{pmatrix} -1 \\ -3+2i \\ -5 \\ -3-2i\end{pmatrix}$ ועתה נוכל להכפיל את הפולינום ע"י הכפלה איבר איבר ונקבל $\begin{pmatrix} 2 \\ 7-9i \\ 0 \\ 7+9i\end{pmatrix}$ ועתה נחזור חזרה לייצוג של מקדמים ונקבל

$$\begin{aligned}
p_0 &\leftrightarrow \begin{pmatrix} -2 \\ 0\end{pmatrix} \\
p_1 &\leftrightarrow \begin{pmatrix} 7-9i \\ 7+9i\end{pmatrix} \\
p_{00} &= -2 \\
p_{01} &= 0 \\
p_0\left(1\right) &= -2 \\
p_0\left(-1\right) &= -2 \\
p_{10} &= 7-9i \\
p_{11} &= 7+9i \\
p_1\left(1\right) &= 14 \\
p_1\left(-1\right) &= -18i \\
p\left(1\right) &= p_0\left(1\right)+p_1\left(1\right) = 12 \\
p\left(-i\right) &= p_0\left(-1\right)-i\cdot p_1\left(-1\right) = -20 \\
p\left(-1\right) &= p_0\left(1\right)-p_1\left(1\right) = -16 \\
p\left(i\right) &= p_0\left(-1\right)+i\cdot p_1\left(-1\right) = 16
\end{aligned}$$

ולכן בסה"כ קיבלנו

$$\begin{aligned} p &\leftrightarrow \frac{1}{4}\begin{pmatrix} 12 \\ -20 \\ -16 \\ 16 \end{pmatrix} = \begin{pmatrix} 3 \\ -5 \\ -4 \\ 4 \end{pmatrix} \\ p(x) &= 4x^3 - 4x^2 - 5x + 3 \end{aligned}$$

ואכן אם נכפיל ישר:

$$\begin{aligned} p(x) &= \left(2x^2 + x - 1\right)(2x - 3) \\ &= 4x^3 + 2x^2 - 2x - 6x^2 - 3x + 3 \\ &= 4x^3 - 4x^2 - 5x + 3 \end{aligned}$$

17.4 בעיה מתרגיל 7

קלט: נתון גרף $G = \langle V, E \rangle$ ו־$\omega : E \to \mathbb{R}^+$ כאשר E מכיל את כל הצלעות ומתקיים א"ש המשולש, כלומר עבור $v, u, t \in V$ מתקיים

$$\omega(u, v) + \omega(v, t) \geq \omega(u, t)$$

המטרה: בהינתם u, v מצא מסילה p מ־u ל־v שעוברת בדרך בכל הקודקודים, כל שמשקל המסילה (סכום הצלעות) מינימלי.

נרצה למצוא אלגוריתם 2 קירוב. (עבור $u = v$ זאת בעיית הסוכן שראינו בהרצאות)

האלגוריתם:

1. נבנה MST.
2. נעבור ב־DFS על הגרף החל מ־u ובשימוש בא"ש המשולש נבנה מסילה העוברת דרך כל הקודקודים.
3. נעקוף את הצלע v באמצעות א"ש המשולש לפי שתי הקודקודים הסמוכים לו (במידה והוא לא האחרון).
4. ונחבר את הקודקוד האחרון עם v.

17.5 פורמט JPEG

טרנספורם פורייה דו־מימדי – בהינתן מטריצה $p(x, y)$ המגדירה $a_{kl}x^ky^l$ ונרצה להעריך את p ב־$\left(\omega_n^k, \omega_n^l\right)$, ולשם כך נגדיר $u_{s,k}(k, l) = e^{\left(\frac{ks}{n} + \frac{lt}{m}\right)2\pi i}$ בדומה להגדרה עבור מימד 1 לפי $u_j^{(k)} = e^{\frac{jk}{n}2\pi i}$.

18 תרגול 10 – 14.12.2010

18.1 זיהוי תבניות – Pattern Matching

נתונים 2 מחרוזות: $s = (s_0, \ldots, s_{n-1})$ ו־$p = (p_0, \ldots, p_{m-1})$ ונניח ש־ $m < n$ כאשר $s_i, p_i \in \{-1, 1\}$.

אנו רוצים למצוא את כל האינדקסים k כך ש־$s_k = p_0, \ldots, s_{k+m-1} = p_{m-1}$.

תזכורת: אם $(a_0,\ldots,a_{n-1})$ ו־$(b_0,\ldots,b_{m-1})$ הם וקטורים, אז הקונבולוציה שלהם היא $(c_0,\ldots,c_{n-1})$ כך שמכפילים וסוכימים את הוקטורים שלהם באופן הבא:

$$\begin{array}{rcl} b_{m-1}\ldots b_1 & b_0 & \\ & a_0 & a_1\ldots \\ \Rightarrow c_0 & = & a_0b_0 \\ b_{m-1}\ldots b_2 & b_1 & b_0 \\ & a_0 & a_1\ldots \\ \Rightarrow c_1 & = & a_0b_1+a_1b_0 \\ & \vdots & \end{array}$$

פתרון: ראשית "נהפוך" את p, ונסמן $\hat{p}=(p_{m-1},\ldots,p_0)$ ואז נבצע $c=s*\hat{p}$.

הטריק: אם $a,b\in\{-1,1\}$ אז $a\cdot b=1$ אם"ם $a=b$, ולכן כשהוקטור $p_0,\ldots,p_{m-1}$ יהיה מעל $s_k,\ldots,s_{k+m-1}$ אז c_{k+m-1} שווה ל־m אם"ם $p_0=s_k,\ldots,p_{m-1}=s_{k+m-1}$, כלומר יש התאמה עד האינדקס ה־$k+m-1$.

18.2 חזרה על אלגברה לינארית

הגדרה 18.1 תהי A מטריצה $n\times m$, אז הדרגה של A מוגדרת ע"י:

1. המספר המקסימלי של עמודות או שורות בת"ל ב־A.
2. המימד של המרחב הוקטורי $Ax\in\mathbb{R}^m$.

מציאת דרגה של מטריצה: דירוג גאוס וספירת מספר השורות ששונות מאפס (לוקח $O\left(m^2n\right)$).

הפיכת מטריצה: A מטריצה $n\times n$, אז A הפכיה אם"ם $rank\left(A\right)=n$.
אפשר להשתמש ב"דירוג גאוס" כדי להפוך מטריצה, כאשר מפעילים כל פעולת שורה גם על מטריצת היחידה - בסיום התהליך כאשר המטריצה A מדורגת קנונית, אז מטריצת היחידה עם הפעולות שהפעלנו תתן את המטריצה ההופכית.

הגדרה 18.2 תהי A מטריצה $n\times n$, אז אם $0\neq v\in\mathbb{R}^n$ מקיים $Av=\lambda v$ אז נאמר ש־v וקטור עצמי (ו"ע) של A עם ערך עצמי (ע"ע) λ.

הערה 18.3 אם למטריצה A יש n ע"ע שונים אז היא שקולה למטריצה אלכסונית D כשהע"ע איברי האלכסון, $A=M^{-1}DM$.

מציאת ערכים עצמיים: מגדירים את $g\left(x\right)=\det\left(A-Ix\right)$ בתור הפולינום האופייני (פ"א) של המטריצה A.
השורשים של הפ"א הם הע"ע של המטריצה (הרי $\left(A-I\lambda\right)v=0$ עבור v ו"ע ו־λ ע"ע).

בעיה: איך מוצאים שורשים של פולינום ממעלה ≤ 5?

פתרון:

1. דרך אחת היא למצוא את שורש ע"י חיפוש בינארי, ואז ניתן לחלק את הפולינום ב־$(x-c)$ (כאשר c הוא שורש) ולחזור על החיפוש בשנית.

2. דרך נוספת היא באמצעות שיטת ניוטון־רפסון: בהינתן נקודה שקרובה דייה (התנאים המדוייקים באינפי2 או וויקיפדיה) לשורש, אז הסדרה הבאה מתכנסת לשורש:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

בעיה: שינוי מאוד קטן במקדמי פולינום, עלול להוביל לשינויים גדולים בשורשי הפולינומים, כלומר השיטה למציאת ע"ע לפי הפולינום היא "לא יציבה חישובית".

הגדרה 18.4 מטריצה A מגודל $n \times n$ תיקרא סימטרית אם $a_{ij} = a_{ji}$ לכל i, j, כלומר $A = A^T$.

עובדות על מטריצות סימטריות:

1. למטריצה סימטרית יש בסיס אורתונורמלי, $v_1, \ldots, v_n$ של ו"ע עם הע"ע $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ (לא בהכרח שונים).
2. A לכסינה ע"י מטריצה אורתוגונלית O (כלומר $OO^T = I$) ומתקיים $A = Q^T DQ$.

18.2.1 שיטת החזקה

אם A מטריצה סימטרית, ואם $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$ הע"ע של A המתאימים לו"ע $v_1, \ldots, v_n$ (מנורמלים), אז נתאר שיטה למציאת λ_1, v_1.

1. יהי x וקטור "כללי", כך ש־ $x = \alpha_1 v_1 + \ldots + \alpha_n v_n$ עבור $\alpha_i \neq 0$ (ניתן למצוא v כזה ע"י בחירה אקראית, ובהסתברות הקרובה ל־1 התנאי יתקיים).
2. בכל שלב $x_{i+1} \leftarrow \frac{Ax_i}{||Ax_i||}$ (ע"מ לדרוש ש־$||x_i|| = 1$)
3. אחרי n איטרציות, נקבל $x_n = z \cdot A^n x_0$ עבור $z \in \mathbb{R}$.

טענה 18.5 $x_n \to v_1$.

הוכחה: $x_0 = \alpha_1 v_1 + \ldots + \alpha_n v_k$ ואז מתקיים

$$\begin{aligned} A^n x_0 &= \alpha_1 A^n v_1 + \ldots + \alpha_n A^n v_n \\ &= \alpha_1 \lambda_1^n v_1 + \ldots + \alpha_n \lambda_n^n v_n \\ &= \lambda_1^n \left(\alpha_1 v_1 + \sum_{i=1}^{n} \overbrace{\left(\overbrace{\frac{\lambda_i}{\lambda_1}}^{<1} \right)^n}^{\to 0} \alpha_i v_i \right) \\ \lim_{n\to\infty} \frac{A^n x_0}{||A^n x_0||} &= \frac{\alpha_1 v_1}{||\alpha_1 v_1||} = \frac{v_1}{||v_1||} = v_! \end{aligned}$$

■

הערה 18.6 אם $\alpha_1 = 0, \alpha_2 \neq 0$, אז בתהליך הנ"ל x_n היה מתכנס ל־v_2 (כי $|\lambda_i| \geq |\lambda_2|$ עבור $i > 2$).

מסקנה 18.7 אם ידוע לנו מיהו v_1, אז אפשר לבחור וקטור כללי x_0, למצוא את הרכיב שלנו בכיוון v_1 (כלומר למצוא $\alpha_1 v_1$ למשל ע"י המכפלה הפנימית, ואז $x_0 \cdot v_1 = \alpha_1$) ולהציב $x_0' = x_0 - \alpha_1 v_1$, ולהפעיל עליו את האלגוריתם בשביל למצוא את v_2.

באותו אופן אפשר להמשיך ולמצוא את $v_3, \ldots, v_n$.

הערה 18.8 כיצד לדעת באיזו איטרציה נעצור ע"מ שנדע שמצאנו קירוב מספיק טוב? נוכל לעקוב אחר סדרת ההפרשים במהלך הריצה, ונעריך את השגיאה בסדר גודל ההפרש.

18.2.2 מטריצה מוגדרת חיובית

הגדרה 18.9 מטריצה סימטרית A מוגדרת חיובית אם אחד מהבאים מתקיים:

1. כל הע"ע שלה חיוביים.
2. לכל $x \in \mathbb{R}^n$ מתקיים כי $0 < x^T A x$.
3. אם $A = B^T \cdot B$ לאיזה מטריצה B (כי $x^T B^T B x = \langle Bx, Bx \rangle > 0$).

18.3 שיטת ה־Least Squares

נתונה A מטריצה $m \times n$ כש־$m > n$ ומתקיים $Ax = b$ (כלומר יש יותר משוואות מנעלמים, ולכן באופן כללי אין פתרון).

המטרה: רוצים למצוא x כך ש־$Ax - b$ יהיה קטן.

שיטת ה־LS מוצאת מינימום ל־$||Ax - b||_2$ וזה שקול למציאת מינימום ל־$||Ax - b||_2^2$.

רעיון: $||Ax - b||_2^2$ היא פונקציה ריבועית בכל אחד מה־x_i־ים. כשנגזור את הפונקציה נקבל n משוואות לינאריות ב־$x_1, \ldots, x_n$, שפתרונן יתן מינימום לפונקציה המקורית.

למה 18.10 מספר טענות עזר על גזירות של וקטורים:

1. $f(\vec{x}) = Bx$ אז $f'(\vec{x}) = B$.
2. $f(\vec{x}) = x^T B x$ אז $f'(\vec{x}) = 2Bx$.

פתרון אלגברי:

$$\begin{aligned} ||Ax - b||^2 &= \langle Ax - b, Ax - b \rangle \\ &= x^T A^T A x - 2x^T A^T b - b^T b \end{aligned}$$

ועתה נגזור ונקבל $2A^T A x - 2A^T b = 0$ ולכן $x = \left(A^T A\right)^{-1} A^T b$.

19 תרגול 11 – לא סוכם

20 תרגול 12 – לא סוכם

21 תרגול 13 – 04.01.2010

21.1 אלגוריתם RSA

האלגוריתם:

1. בוחרים 2 ראשוניים p, q גדולים באמצעות אלגוריתם מילר רבין (בהמשך התרגול).
2. נחשב את $n = pq$ ו־$\phi(n) = (p-1)(q-1)$ שהוא גודל החבורה בה נעבוד.
3. בוחרים $e \in \mathbb{Z}^*_{\phi(n)}$, ומחשבים את $d = e^{-1}$ בתוך החבורה.
4. המפתח הציבורי הוא (n, e) והמפתח הסודי הוא (n, d).
5. הודעה הוא מספר m ב־$\mathbb{Z}^*_n$ ואז הקידוד הוא $m \mapsto m^e \pmod n$ (באופן כללי בשביל לחשב את a^n ב־$\log n$ נבצע את החישוב ע"י $a, a^2, a^4, a^8, \ldots$ עד $a^{\lfloor \log_2 n \rfloor}$ כאשר אם נציב את n ע"י סכום של חזקות 2 (ייצוג בינארי) אז אם $n = 2^{m_1} + 2^{m_2} + \ldots + 2^{m_k}$ אז נוכל לחשב $a^n = a^{2^{m_1}} \cdots a^{2^{m_k}}$).
6. הפענוח הוא $(m^e)^d$ ולכן בשימוש במשפט פרמה מתקיים

$$\begin{aligned} m^{ed} &= m \cdot \left(\overbrace{m^{\phi(n)}}^{=1}\right)^k \equiv_n m \\ ed &= 1 + k \cdot \phi(n) \end{aligned}$$

21.2 זמן ריצה של GCD

תזכורת: חוזרים על התהליך עבור $a > b$ כך ש־ $(a, b) \mapsto (b, a\%b)$ עד אשר אחד האיברים שווה ל־0 כאשר

$$\begin{aligned} a &= kb + q \\ a\%b &= q \\ a &\geq b + q \Rightarrow q < \frac{a}{2} \end{aligned}$$

מספר השלבים בריצת האלגוריתם הוא $\log n$.
פעולת החילוק עבור מספר באורך m במספר באורך n היא $O(\log m \cdot \log n)$ ובסה"כ $O\left((\log n)^3\right)$.

21.3 מילר רבין

עובדות:

1. אם n ראשוני אז לכל $1 < a < n$ מתקיים כי $a^{n-1} \equiv_n 1$.

2. אם n ראשוני, אז ל־$x^2 \equiv_n 1$ יש בדיוק 2 פתרונות אפשריים: 1 ו־-1 (בשדה $\mathbb{Z}_n^*$ זהו $n-1$ ו־1).

האלגוריתם:

1. נבדוק ש־n לא זוגי.
2. נגריל מספר $1 < a < n$.
3. נבדוק האם $a^{n-1} \equiv_n 1$ ואם לא נחזיר "פריק" (בשימוש בעובדה 1).
4. נרשום $n-1 = u \cdot 2^s$ כאשר u אי זוגי ו־$s > 0$.
נתבונן ב־ $a^u, a^{2u}, a^{4u}, \ldots, a^{2^s u}$ (האחרון כמובן שווה לאחד) כאשר נשים לב שכל מספר הוא ריבוע של המספר הקודם, ולכן אם $a^{2^j u} = 1$ אז גם כל הבאים אחריו, ולכן נמצא את המקום הראשון שבו $a^{2^j u} = 1$.
אם המקום הקודם לו שונה מ־-1 אז נחזיר פריק (בשימוש בעובדה 2).
5. נחזיר n כנראה ראשוני.

טענה 21.1 יהי n פריק, אז בהסתברות $\leq \frac{1}{2}$ האלגוריתם יחזיר פריק.

הוכחה: צריך להראות שבהסתברות $\leq \frac{1}{2}$ מתקיים כי n נכשל במבחן הראשון או בשני.
כדי להראות שבהסתברות $\leq \frac{1}{2}$ מתקיים ש־n נכשל במבחן, מספיק להראות שלפחות חצי מה־a־ים גורמים לו להיכשל.

1. אם n לא מספר Carmicheal (להבא CM) אז יש $a \in \mathbb{Z}_n^*$ כך ש־$a^{n-1} \neq 1$.
נגדיר $B = \left\{x \in \mathbb{Z}_n^* : x^{n-1} = 1\right\}$ ומתקיים כי $B \lneq \mathbb{Z}_n^*$ כי n לא CM.
   (א) B תת חבורה כי $a, b \in \mathbb{B}$ אז $1 = a^{n-1}b^{n-1} = (a \cdot b)^{n-1}$ ולכן $ab \in B$.
   אם $a \in B$ אז $\left(a^{-1}\right)^{n-1} = \left(a^{n-1}\right)^{-1} = 1^{-1} = 1$ ולכן $a^{-1} \in B$ ולכן זאת תת חבורה.
   (ב) כיוון ש־B תת חבורה של $\mathbb{Z}_n^*$, אז $|B| \mid |\mathbb{Z}_n^*| = \phi(n) \leq n-1$ ולכן $|B| \leq \frac{n}{2}$.
2. אם n הוא CM. נסמן $\left(\mathbb{Z}_n^*\right)^k = \left\{x^k | x \in \mathbb{Z}_n^*\right\}$.
ידוע לנו ש־$\left(\mathbb{Z}_n^*\right)^{n-1} = \{1\}$ ולכן נסתכל על המבחן השני ומתקיים $n-1 = 2^s \cdot u$.
יהי j האינדקס המקסימלי כך ש־$\left(\mathbb{Z}_n^*\right)^{2^j \cdot u} \neq \{1\}$ וקיים j כזי כי $\left(\mathbb{Z}_n^*\right)^u \neq \{1\}$ כי $(-1)^u = -1$ ולכן j מוגדר היטב.
נגדיר $B = \left\{a \in \mathbb{Z}_n^* | a^{2^j \cdot u} = 1, -1\right\}$.
נניח ש־$a \notin B$, אז $a^{2^j u} \notin \{1, -1\}$, אבל $\left(\mathbb{Z}_n^*\right)^{2^{j+1} u} = \{1\}$ ולכן $a^{2^{j+1} u} = 1$.
אז a יגרום ל־B להיכשל במבחן השני.
   (א) נראה ש־B תת חבורה:
   i. אם $a, b \in B$ אז

$$(ab)^{2^j u} = \overbrace{a^{2^j u}}^{\in\{1,-1\}} \overbrace{b^{2^j u}}^{\in\{1,-1\}} \in \{1, -1\} \Rightarrow ab \in B$$

   ii. אם $a \in B$ אז

$$\left(a^{-1}\right)^{2^j u} = \left(\overbrace{a^{2^j u}}^{\in\{-1,1\}}\right)^{-1} \in \{-1, 1\}$$

כי $1^{-1} = 1$ ו־$-1 = (-1)^{-1}$.

(ב) נראה ש־$B \lneq \mathbb{Z}_n^*$ (כלומר תת חבורה ממש).

i. עובדה: בחבורה $\mathbb{Z}_{p^e}^*$ עבור p ראשוני, יש איבר g מסדר $\phi(n) = p^e\left(1-\frac{1}{p}\right)$

ii. אם n הוא CM, אז n לא חזקה של ראשוני. נניח שכן, אז יהי g איבר מסדר $\phi(n)$ ב־$\mathbb{Z}_n^*$ ולכן

$$\begin{aligned} g^{p^e-p^{e-1}} &= 1 \\ g^{n-1} = g^{p^e-1} &= 1 \\ \Rightarrow g^{(p^e-1)-\left(p^e-p^{e-1}\right)} = g^{p^{e-1}-1} &= 1 \end{aligned}$$

וזו בסתירה לכך ש־$ord(g) = p^e - p^{e-1} > p^e - 1$.

iii. אפשר לרשום $n = n_1 \cdot n_2$ כש־n_1 ו־n_2 זרים.
יש איבר v כך ש־$v^{2^j u} \neq 1$. אם $v^{2^j u} \neq -1$ אז $v \notin B$ וסיימנו.
אחרת $v^{2^j \cdot u} = -1$, ונגדיר $r_1 := v \bmod n_1$ ו־$r_2 := v \bmod n_2$, ואז מתקיים כי $r_1^{2^j \cdot u} \equiv_{n_1} -1$ ו־$r_2^{2^j u} \equiv_{n_2} -1$.
נבנה מספר x כך ש־$x^{2^j u} \equiv_{n_1} 1$ ו־$x^{2^j u} \equiv_{n_2} -1$ ו־x כזה לא יכול להיות ב־B.
לפי משפט השאריות הסיני יש x כך ש־

$$\begin{aligned} x \bmod n_1 &= 1 \\ x \bmod n_2 &= r_2 \end{aligned}$$

ולכן $x^{2^j u} \equiv_{n_1} 1$ ו־$r_2^{2^j u} \equiv_{n_2} -1$ ולכן $x^{2^j u} \equiv_{n_2}$ ולכן $x^{2^j u} \notin \{-1,1\}$ ולכן $x \notin B$ ולכן B תת חבורה ממש ולכן $|B| \leq \frac{|\mathbb{Z}_n^*|}{2} \leq \frac{n-1}{2}$.

■

22 תרגול 14 – 11.01.2011

22.1 פיצוח ה־RSA

תזכורת: יש 2 ראשוניים גדולים p, q ו־$n = p \cdot q$ ונתון המפתח הציבורי (n, e).

אם נתון $d = e^{-1}$ בחבורה $\mathbb{Z}_{\phi(n)}^*$ אז נרצה לפקטר את n. קיים אלגוריתם הסתברות המסוגל לפרק את n לגורמים.

נשים לב ש־$e \cdot d \equiv_{\phi(n)} 1$ ולכן $ed = k\phi(n) + 1$ ונציג אותו עבור u אי זוגי

$$ed - 1 = 2^s \cdot u$$

ונשים לב ש־$a \in \mathbb{Z}_n^*$ אז מתקיים כי $a^{2^s \cdot u} \equiv 1$.

האלגוריתם:

1. נגריל מספר a מ־$\mathbb{Z}_n^*$.
2. נחשב את $a^u, a^{2u}, \ldots, a^{2^s u}$.
3. יהי j האינדקס המקסימלי בו $a^{2^j u} \neq 1$ (בתנאי שקיים כזה), אז אם $a^{2^j \cdot u} \neq -1, 1$ אז מצאנו שורש לא טריוויאלי של 1 ונסמנו ב־x.

4. כעת $gcd\left(x+1,n\right)$ הוא מחלק $1 \neq$ של n, גלומר או p או q).

הערה 22.1 נשים לב ש־$x^2 =_n 1$ ולכן $x^2-1 \equiv_n 0$ ולכן $\left(x+1\right)\left(x-1\right) = kn$ ולכן $n \mid \left(x+1\right)\left(x-1\right)$ ולכן $gcd\left(x+1,n\right) \neq 1$.

שאלה: מה ההסתברות שמצאנו a "טוב"?

תשובה: לפחות $\frac{1}{2}$, כי נראה שה־a־ים ה"רעים" מוכלים בתת חבורה ממש של $\mathbb{Z}_n^*$ (כמו במילר־רבין).
נגדיר את j להיות האינסקס המקסימלי כך ש־$\left(\mathbb{Z}_n^*\right)^{2^j \cdot u} \neq \{1\}$, ונגדיר

$$B = \left\{a \in \mathbb{Z}_n^* | a^{2^j \cdot u} \in \{-1,1\}\right\}$$

מטיעון דומה כמו במילר רבין ניתן להראות ש־B היא תת חבורה ממש.
אנו יודעים שיש v כך ש־$v^{2^j \cdot u} \neq 1$ מהגדרת j, ונניח ש־$v^{2^j u} =_n -1$, אז $v^{2^j u} =_p -1$ וגם $v^{2^j u} =_q -1$.
לפי משפט השאריות הסיני קיים x כך ש־$x =_p v$ ו־$x =_q 1$ ואז $x^{2^j u} =_p -1$ ו־$x^{2^j u} =_q 1$ ולכן $x^{2^j u} \neq_n \pm 1$,
ולכן $x \notin B$ ולכן היא תת חבורה ממש.

22.2 המבחן

22.2.1 מבנה המבחן

חלק א': 5 שאלות קצרות, כאשר כל אחת שווה ל־5 נקודות.

חלק ב': 5 שאלות ארוכות, בחירה של 4 מתוך 5.

22.2.2 שאלות ממבחנים קודמים

שאלה: נתונות m מחרוזות $w_1,\ldots,w_m$ מעל א"ב Σ, ונתונה מחרוזת u. רוצים להציג את u בתור שרשור של מחרוזות מ־$w_1,\ldots,w_m$ (עם חזרות).
השרשור v חייב להיות שווה לאורך של u, אבל ייתכנו אי שוויונות בתווים.
המחיר של החלפת $x \in \Sigma$ ב־$y \in \Sigma$ הוא $c\left(x,y\right)$.
לדוגמא אם

u	a	b	a	c	b
v	a	b	c	c	a

המחיר הוא

$$c\left(a,a\right)+c\left(b,b\right)+c\left(a,c\right)+c\left(c,c\right)+c\left(b,a\right)$$

רעיון לפתרון: באופן דומה ל־Knapsack נתבונן במחרוזת האחרונה בשרשור ונראה מה קורה כשמסירים אותה.
עבור s מחרוזת חלקי של u, s^k היא הרישא של k התווים הראשונים של s, s_j הוא התו ה־j של s ו־$l\left(s\right)$ הוא האורך של s ואז מתקיים

$$\begin{aligned} f\left(s\right) &= \min_{\substack{1 \leq i \leq n \\ l\left(w_i\right) \leq l\left(s\right)}} \left(f\left(s^{l(s)-l(w_i)}\right) + \sum_{j=1}^{l(w_i)} c\left(s_{l(s)-l(w_i)+j}, \left(w_i\right)_j\right) \right) \\ f\left(s^0\right) &= 0 \end{aligned}$$

תתי הבעיות הם הרישות של u ויש $l(u)$ כאלה, וכל תת בעיה דורשת $\sum_{i=1}^{m} l(w_i)$ ולכן זמן הריצה הוא $l(u)\sum_{i=1}^{m} l(w_i)$.

פתרון מלא:

האלגוריתם: (למשל) עבור k שרץ מ־1 עד $l(u)$ נחשב את $f(u^k) = \min(\ldots)$, אם w_i נותן מינימום לביטוי אז נרשום $p_k = i$.
אם $f(u_{f(u)}) = \infty$ נחזיר אין פתרון.
אחרת, נבנה את v באופן ריקורסיבי (כאשר + משרשר מחרוזות)

$$v\left(u^k\right) = v\left(u^{k-l\left(w_{p_k}\right)}\right) + w_{p_k}$$

ונחזיר את $v(u)$.

זמן ריצה: כמו שנכתב לעיל.

נכונות: נניח ש־v הוא פתרון אופטימלי עבור u, אז נסמן את המחרוזת האחרונה בשרשור u בתור w_j.
נראה באינדוקציה ש־$f(u^k)$ הוא המחיר המינימלי של בניית u^k כשרשור של $w_1, \ldots, w_m$.
עבור $k = 0$ זה ברור.
נניח שיש פתרון טוב יותר מ־$f(u^k)$ ונסמנו ב־v' ו־v' מסתיים ב־w'_j ולכן

$$\sum_{q=1}^{l(w_j)} c\left(u_{k-l\left(w'_j\right)+q}, \left(w'_j\right)_q\right) + c\left(u_{k-l(w_j)}\right) = c(u_k, v)$$

ומהנחת האינדוקציה אנו ידועים ש־

$$\begin{aligned}
\sum_{q=1}^{l(w_j)} c\left(u_{k-l\left(w'_j\right)+q}, \left(w'_j\right)_q\right) + f\left(u_{k-l\left(w'_j\right)}\right) &\leq c(u_k, v) \\
\min_{\substack{1 \leq i \leq n \\ l(w_i) < l(w_j)}} f\left(u_{k-l(w_j)}\right) + \sum () &\leq
\end{aligned}$$

ולכן סתירה להנחה.
נותר להראות כי האלגוריתם אכן נותן פתרון חוקי.

שאלה: יש n מרגלים הנמצאים ב־n ערים באסיה, ונרצה להעביר אותם ל־n ערים באמריקה, כאשר כל המסלולים עוברים דרך טיסת חיבור ב־n ערים באירופה, ובכל עיר יכול להיות בכל רגע נתון מרגל אחד לכל היותר.

פתרון: באמצעות רשת זרימה. נגדיר n קודקודים As_i, n קודקודים Am_i, ו־n קודקודים Es_i, Ef_i, וקודקוד מקור s וכיור t.
נגדיר צלע בין s ל־As_i, צלע בין As_i ל־Es_j אם קיימת טיסה בינם, בין Es_i ל־Ef_i, בין Ef_i ל־Am_j אם יש טיסה בינהם ובין Am_i ל־t, ולכל הצלעות ניתן משקל של 1.
זמן הריצה הוא $O(n^3)$ כי יהיו לכל היותר n שלבים באלגוריתם EK וה־BFS בכל שלב דורש $|E| + |V|$ ולכן בסה"כ $O(n^3)$ (חשוב לשים לב לכך שבשימוש בזמן הריצה הרגיל של EK היינו מקבלים $O\left(|V||E|^2\right) =$ $O(n^5)$).