אלגוריתמים – דף נוסחאות

1 כללי

טענה: כל 2 מהתכונות הבאות גוררת את השלישית: (1) גרף קשיר (2) חסר מעגלים (3) בעל |V|-1 קשתות.
(*) בעץ בין כל זוג צמתים קיים מסלול אחד ויחיד. (*) בגרף עם מעגל אוילר אין גשר.
מעגל אוילר - גרף <u>לא מכוון וקשיר</u> מכיל מעגל אוילר ⟺ כל הדרגות זוגיות. גרף <u>מכוון וקשיר</u> מכיל מעגל אוילר ⟺ לכל v ב-V: din(v)=dout(v).
מסלול אוילר - גרף <u>לא מכוון וקשיר</u> מכיל מסלול אוילר ⟺ בגרף 2 קודקודים מדרגה אי-זוגית והיתר מדרגה זוגית. גרף <u>מכוון וקשיר</u> מכיל מסלול אוילר ⟺ קיימים 2 צמתים w,v כך ש-din(w)+1=dout(w), din(v)=dout(v)+1 וגם לכל שאר הקודקודים u≠v,w מתקיים din(u)=dout(u).
גשרים - קשת אינה גשר ⟺ היא נמצאת על מעגל פשוט בגרף. גרף <u>לא מכוון וקשיר</u> ניתן לכיוון לגרף קשיר בחוזקה ⟺ אין בו גשרים.

2 BFS

טענה: אם u נגיש מ-s, BFS מבקר ב-u. (הוכחה באינדוקציה).
טענה: בכל רגע, בתור Q נמצאים צמתים עם לכל היותר 2 ערכי d עוקבים בסדר עולה. כלומר, יש בתור ערך d אחד לכל הצמתים או 2 ערכי d עוקבים בלבד). (הוכחה באינדוקציה).
משפט: לכל צומת u, d[u]=δ(s,u). **הוכחה:** צמתים עם ערך אינסוף נוכיח כמעט ישירות ושאר הצמתים באינדוקציה.
(*) בגרף <u>מכוון</u>, כל קשת שה-BFS לא השתמש בה לבניית d, מחברת צומת בשכבה j כלשהי לצומת בשכבה i≤j+1. בגרף <u>לא מכוון</u> כל קשת מחברת צמתים באותה שכבה או בשכבות סמוכות.
טענה: גרף <u>לא מכוון</u> הוא **דו"צ** ⟺ אין בו מעגל באורך אי זוגי (כלומר בBFS אין 2 צמתים באותה השכבה משני צדדים שונים).
(כדי לחלק את הגרף לקבוצות, נגדיר v1 להיות כל הצמתים בשכבות הזוגיות ו-v2 בשכבות האי זוגיות)
טענה: לכל קשת (u,v) מתקיים בסיום האלגוריתם: δ(s,v) <= δ(s,u)+1 (אם (u,v) במק"ב מ-S ל-V יש שיוויון).

3 DFS

שימושים: (1) מציאת רכיב קשירות בגרף <u>לא מכוון</u> - נריץ DFS, כל עץ בפלט שלו הוא רכיב קשירות. **(2) האם 2 צמתים u,v נמצאים באותו רכיב קשירות?** - אם הם באותו עץ אחרי הDFS אז כן. נבדוק עץ ע"י הליכה אחורה עם מצביעי π ובדיקה שיש אותו אב. **(3) האם גרף נתון מכיל מעגל?** קיים מעגל <u>אמ"מ</u> קיימת קשת אחורית. **(4) מיון טופולוגי של גרף מכוון א-ציקלי (DAG)** – מסדר את הצמתים בסדר V1,…,Vn כך שלכל קשת (Ui,Uj)מתקיים i<j. כדי למצוא, נריץ DFS וסדר הצמתים יהיה הפוך לערכי f(·) (V1 עם ערך f מקסימלי). **(5) מציאת גשרים, קודקודים מנתקים ורכיבי קשירות** – גרף <u>לא מכוון</u> וקשיר ניתן לכוון לגרף קשיר בחוזקה ⟺ אין בו גשרים. **(6) מציאת רכיבי קשירות חזקה <u>בגרף מכוון</u>** - (1) נריץ DFS על G ונסדר את הצמתים לפי סדר f יורד ונקרא לרשימה L. (2) נבנה את הגרף ההפוך G^T. (3) נריץ DFS על GT, כאשר הפרוצ' הראשית משתמשת ב-L בתור רשימת הצמתים. (4) כל עץ בDFS השני הוא רק"ח. (*) גרף העל הוא DAG.
(*) כדי לבדוק אם גרף הוא **קשיר חזק**, נריץ DFS מ-s על G ועל G^T. אם בשני המקרים קיבלנו עץ יחיד, G קשיר חזק.
עקרון הקינון - לכל צומת u מתאים מרוח זמן חי [d(u),f(u)]. ל-2 צמתים,יש להם מרוחים זרים או מרוחים מקוננים זה בזה.
עקרון המסלול הלבן – נניח ב G-יש מסלול V0→V1→…→Vn ונניח שהקודקוד הראשון במסלול שהDFS מקלה הוא V0. אזי כל שאר הצמתים יהיו צאצאים של V0 ב-DFS. (הוכחה באינדוקציה).
(*) בגרף <u>לא מכוון</u> אין בעץ הDFS קשתות קדמיות או קשתות חוצות אלא רק קשתות עץ וקשתות אחוריות.
טענה: לכל 2 רכיבי **דו קשירות** יכול להיות רק **קודקוד משותף** 1. **מסקנה:** רכיבי דו"ק מהווים חלוקה של הקשתות לקב' זרות.
(*) ה-DFS עובר על כל קשתות הגרף, בחלק מהן הוא משתמש להגדרת π, לבניית עץ הDFS, ובשאר לא. לקשתות המשמשות להגדרת π נקרא קשתות עץ. האחרות מתחלקות ל: (1) קשת קדמית- מחברת אב קדמון לצאצא. (2) קשת אחורית- מצאצא לאב קדמון (או לאב עצמו).
(3) קשת חוצה- מחברת 2 צמתים שאינם צאצא/אב קדמון. (*) בגרף <u>מכוון</u>, מסתכלים על קשת (U,V) ברגע ש-U בודק אותה: (1) V לבן ← קשת עץ.
(2) אפור←אחורית. (3) שחור←קדמית/חוצה. (*) בגרף לא מכוון יש רק 2 סוגי קשתות – עץ ואחורית.

4 עץ פורש מינימאלי (עפ"מ)

קלט: גרף <u>לא מכוון</u>. **תכונות:** (1) פורש את הגרף. (2) סכום המשקולות של הקשתות בעץ הוא המינימאלי מבין הפורשים.

קרוסקל
מציאת עפ"מ באמצעות האלגוריתם של קרוסקל – בכל שלב מחפשים קשת עם משקל מינימאלי, המחברת 2 רכיבי קשירות ב-A. הקשתות ממוינות ע"פ משקל בסדר עולה. **סבוכיות:** O(|E|log|E|)=O(|E|log|V|).
(*) הפלט של האלגוריתם של קרוסקל תלוי אך ורק בסדר הממוין של הקשתות.
(*) אם לכל הקשתות יש משקלים שונים אז אלגוריתם קרוסקל יוצר עץ אחד יחיד.
(*) האלגוריתם של קרוסקל מסוגל לייצר כל עפ"מ אפשרי. **מסקנה:** אם המשקלים שונים, יש עפ"מ יחיד.
אם ניתן לבצע מיון בזמן ליניארי (משקל חסום), הסיבוכיות – O(|E|α(|E|,|V|)).

Prim
הרעיון הוא לתחזק רכיב קשירות אחד C ובכל פעם להוסיף לו צומת. האלגוריתם משתמש בתור עם עדיפות כדי לתחזק את הקודקודים. בכל שלב אנו שומרים עבור כל קודקוד את המשקל המינימאלי המחבר אותו אל C.
סבוכיות: בערימה רגילה – O(|E|log|v|) (כמו קרוסקל). בערימת פיבונאצ'י – O(|V|log|V| + |E|). במערך לא ממוין העלות היא O(|V|²+|E|) (אם $|E|=O(|V|^2)$ אז העלות היא למעצה O(|E|).

טענה: תהי f:ℛ⟶ℛ פונקציה עולה ממ, אם נגדיר משקל חדש w*(e)=f(w(e)) T ז עפ"מ לפי w ⟺ T עפ"מ לפי *w.
טענה: T_1,T_2 עפ"מים באותו גרף עם אותה w, אז אם נסדר את משקלי הקשתות בצורה ממוינת נקבל כי x1=y1, x2=y2… .

5 מסלולים קצרים ביותר (מק"ב)

מסלול קצר ביותר מצומת s לכל צומת אחרת בגרף (בהתחשב במשקל על הקשתות). מניחים שאין מעגלים שליליים.
תכונות מק"בים: (1) תת מסלול של מק"ב הוא מק"ב. (2) **אי שוויון המשולש:** אם (u,v)∈E אז δ(s,v)≤δ(s,u)+w(u,v). (3) אם u צומת אחד לפני v על מק"ב מ-s אל v אז δ(s,v)=δ(s,u)+w(u,v). (4) אם כל המשקולות הן 1, אז משקל מסלול=מס' הקשתות ונפתור באמצעות BFS.

דייקסטרה
דייקסטרה מוצא מק"בים כאשר המשקולות הן **אי שליליות**. **סבוכיות:** ערימה רגילה - O(|V|log|V|+|E|log|E|), ערימת פיבונאצ'י – O(|E|+|V|log|V|). באופן כללי – O(|V|+|E|+|V|*extract_min+|E|*update). (*) אם למשקל יש חסם, ניתן למש את הADT של הערימה באמצעות מערך ולקבל סבוכיות של O(|E|+|V|).

טופולוגי
טופולוגי - (נריץ למציאת מק"ב) אם הגרף הוא א-ציקלי, ניתן לבצע מיון טופולוגי (DFS ב-O(|E|+|V|)) ולרוץ על ערכי הצמתים אחד אחד ולכל צומת לעדכן את כל הקשתות היוצאות ממנו ע"פ התנאי d[t]=min{d[t],d[v]+w(u,t)}. **סבוכיות:** זמן – O(|E|+|V|) מקום – O(|V|).

פורד – קלט: גרף כללי, משקולות <u>אולי</u> שליליים, <u>אין מעגלים שליליים</u>. זמן ריצה איום ונורא. מפורט בחוברת.

BF
בלמן פורד (BF) – בכל איטרציה עובר על כל הקשתות ומבצע Relax. פועל על גרף מכוון ומשוקלל. אם יש מעגל שלילי, מזהה ומחזיר שגיאה.
סבוכיות: O(|V||E|). מופיע בחוברת.

בעת ריצת האלגוריתם למציאת מק"ב, בכל שלב של האלגוריתם מצביעי π שאינם NIL + השורש S, יוצרים עץ מכוון לכיוון S (בהנחה שאין מעגלים שליליים). צמתי העץ יהיו אלו הנגישים מ-s. לכל צומת u בעץ המסלול מ-s אליו הוא מק"ב.
<u>אלגוריתם לקבלת גרף-המק"בים מעץ-המק"בים</u>: מוסיפים לעץ כל קשת המקיימת: $d(v) = d(u) + w(u, v)$.
<u>ספירת מספר מסלולים</u>: בעזרת תכנות דינאמי, נגדיר n(v) מספר המסלולים מ-s ל-v. n(s)=1, $n(u)=\sum_{(v,u)\in E} n(v)$, ב-O(n) (ניתן להכליל...).

6 All-Pairs Shortest Paths

BF ניקוי
מחקה את BF באופן מטריצאלי. במקום לבצע חישוב ישיר, מבצעים "כפל" של מטריצות, ע"י החלפת פעולת החיבור בפעולת min ופעולת כפל בפעולת החיבור. הסיבה שניתן לעשות את השינוי היא כיוון שפעולת min היא קומוטטיבית אסוציאטיבית.**סבוכיות:** במקור $O(|V|^4)$ אבל את הכפלת המטריצות ניתן לבצע log|v| פעמים במקום |V| [במקום (..w*w)*w)*w)…)) נכפיל (..w*w)(w*w)(w*w)..)] ונקבל $O(|V|^3\log|V|)$.

FW
Floyd-Warshall – בכל שלב מנסים לבדוק האם מסלול "בגובה" k, קצר יותר ממסלול שחושב עד עתה (עד גובה k-1). אם כן, מעדכנים. אחרת, לא. **סבוכיות:** $O(|V|^3)$. **הערות:** (1) **זיכרון** האלגוריתם מייצר n+1 מטריצות. בפועל מספיקות 2 ובעצם רק 1. (2) האלגוריתם שראינו מחשב רק אורך. כדי לחשב את המסלול צריך לעדכן גם מטריצת π. אם שונה $\pi_{ij}^{(k)} \leftarrow \pi_{kj}^{(k-1)}$. אחרת $\pi_{ij}^{(k)} \leftarrow \pi_{ij}^{(k-1)}$.

ג'ונסון
Johnson - הופך את המשקולות לאי שליליים ומריץ דייקסטרה v פעמים. מתבסס על הלמה. **סבוכיות:** $O(|V|^2\log|V|+|V||E|)$.
משתמש ברשימת שכנויות כדי לייצג את הגרף. כדי לחשב ערך h מוסיפים צומת s, מוריצים BF ואז בוחרים h(v)=δ(s,v) לכל v.

! למה: תהא h:V-ℝ h פונ' על הצמתים ונגדיר משקולות חדשים כך w*(u,v)=w(u,v)+h(u)-h(v) לכל קשת אז מק"ב תחת w⟺מק"ב תחת *w.

7 זרימה ברשתות

תכונות זרימה חוקית:(1) אילוצי קיבול f(u,v)≤c(u,v). (2) אנטי סימטריה f(u,v)=-f(v,u). (3) שימור זרימה: לכל צומת u≠s,t $\sum_v f(u,v) = 0$.
זרימה בין קבוצות: מוגדרת להיות הזרימה מכל צומת בקבוצה A לכל צומת בקבוצה B. **תכונות:** (1) f(A,B)=-f(B,A). (2) אם X,Y זרות $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$ וכן $f(Z, X \cup Y) = f(X,Z) + f(Y,Z)$. **הרשת השיורית** – עותק של G בו מוגדר $c_f(u,v)=c(u,v)-f(u,v)$.

פורד פלקרסון
פורד פלקרסון – כל עוד קיים מסלול משפר, נגדיל את f ע"י הזרמה נוספת לאורך המסלול. מסלול משפר הוא מסלול מ-s אל t ב-G_f (אם הקיבול השיורי הוא 0 לא יופיע ב-G_f). $C_f(p)$ הוא הקיבול המינימאלי מבין הקשתות במסלול משפר p. **סבוכיות:** תלוי <u>בקיבולים</u>. כאשר הזרימה היא בשלמים, יקח לכל היותר צעדים השווים לזרימה המקסימלית. הרעיון הוא לחפש מסלולים משפרים קצרים <u>ביותר</u> מבחינת מספר הקשתות. כך חוסמים את מס' האיטרציות במספר המסלולים המשפרים. עלות האלגוריתם תהיה איטרציות כפול עלות BFS. כלומר, $O(|E|^2|V|)$ (מס' איטרציות הוא O(|E||V|) והגרף קשיר לכן עלות BFS היא O(|E|)).

MinCut-MaxFlow – הזרימה המקסימלית שווה לקיבול החתך המינימאלי. ע"פ המשפט, התנאים הבאים שקולים: (1) f זרימה מקסימאלית. (2) הרשת השיורית G_f לא מכילה מסלול מ-s אל t. (3) |f|=c(S,T) לאיזשהו חתך (S,T).
טענה: לכל זרימה f ולכל חתך (S,T) מתקיים |f|=f(S,T). **טענה:** לכל זרימה f ולכל חתך (S,T): |f|=f(S,T)≤c(S,T)).

אדמונדס-קארפ
אדמונדס-קארפ - נאתחל את הזרימה לזרימת אפס. נבנה רשת שיורית. נריץ BFS על הרשת ונחפש מסלול קצר בין S ל T. נשפר את הזרימה לפי המקסימום שאפשר להזרים במסלול המשפר שמצאנו. נחזור על התהליך (בניית רשת שיורית מחדש וכולי). **סבוכיות:** עלות של BFS כפול מספר האיטרציות. עלות BFS היא O(|E|) כי הגרף קשיר ומספר האיטרציות חסום ע"י O(|E||V|) ולסיכום העלות היא $O(|E|^2|V|)$.

דיניץ'
דיניץ' – הרעיון הוא לבנות רשת שכבתית מהרשת השיורית (ע"י BFS), לחפש בה כמה שיותר מסלולים משפרים ורק כאשר t לא נגיש יותר, לבנות רשת שיורית חדשה. שיטת איתור המסלולים ברשת השכבתית דומה ל-DFS. כאשר "מסיימים" עם צומת מוחקים אותו וכל הקשתות הנכנסות אליו. כאשר מגיעים ל-t מזרימים זרימה מקסימאלית במסלול, מוחקים קשתות רוויות וממשיכים להתקדם מהאב של הקשת הרוויה הקרובה ביותר לצומת s. **סבוכיות:** $O(|E||V|^2)$ (בכל פאזה לכל היותר O(|E||V|) צעדים כפול מס' פאזות השכבות המקסימליות (נגזר ממספר השכבות המקסימאלי) – O(|V|)).
ראה/י ברשתות 1/0 עבור סבוכיות טובה יותר.

רשתות 0/1
קיבול כל קשת היא 1 (או 0). דיניץ מאוד יעיל ברשתות אלו. ברשתות כאלו כל פאזה של דיניץ' מתבצעת ב-O(|E|) (כי במחיקה כל הקשתות במסלול רוויות). לכן, ריצת דיניץ כולה לוקחת O(|E||V|). לעיתים ניתן למצוא חסם הדוק יותר ע"פ טיפוס.
<u>רשת טיפוס 1</u> – רשת בה אין קשתות מקבילות (או אנטי מקבילות). **סבוכיות:** $O(|E|*\min(|E|^{1/2},|V|^{2/3}))$. <u>רשת טיפוס 2</u> – רשת בה לכל צומת יש דרגת כניסה 1 או דרגת יציאה 1. **סבוכיות:** $O(|E||V|^{1/2})$. (*) טיפוס הרשת (1 או 2) נשמר גם ברשת השיורית! (השיפור מושג ע"י שיפור החסם על מספר הפאזות הדרושות).

טענה: נניח זרימה נוכחית היא 0 וערך זרימה מקסימאלית M. אזי מספר השכבות הוא $k \le \frac{|E|}{M}$ ברשת כללית ו- $k \le \frac{|E|-2}{M} + 1$ בטיפוס 2.

זווג מושלם
משפט Hall – יהיה G גרף דו"צ $V = X \overset{\cup}{\tau} Y$,E⊆XxY ,|X|=|Y|. לכל A⊆X נסמן ב-Γ(A) את קבוצת השכנים של צמתים של A ב-Y. אז ל-G יש זיווג <u>מושלם</u> (כלומר זיווג בגודל |X|=|Y|) אמ"מ לכל A⊆X מתקיים |Γ(A)|≥|A| .

משפט מנגר – G=(V,E) גרף מכוון. s,t∈V. מספר הקשתות שצריך למחוק כדי לנתק את s מ-t שווה למספר המסלולים הזרים.
<u>**שימושים**</u> – <u>מציאת זיווג מקסימלי בגרף דו"צ</u>, <u>מספר מקסימלי של מסלולים זרים בקשתות\קודקודים</u>, <u>**k-קשירות:**</u> גרף נקרא k-קשיר בקשתות אמ"מ יש לפחות k מסלולים זרים בקשתות (לכל) מ-u ל-v. אלגוריתם – נקבע u צומת כלשהו, ונבדוק לכל צומת v האם יש k מסלולים זרים בקשתות (בעזרת זרימה) מ-u ל-v ובכיוון הנגדי ⟺ הגרף k-קשיר בקשתות. סיבוכיות - $O(|V||E|\min(|E|^{\frac{1}{2}}, |V|^{\frac{2}{3}}, k))$.

8 התאמת מחרוזות

בשיטה נאיבית נוכל לבצע השוואה מכל מקום במחרוזת T. זה מבוצע בזמן O(mn). ראינו שיטת השוואות באמצעות טבלת δ שמבצעת את ההשוואה בזמן O(m+n). כמו כן, ראינו שיפור נוסף באמצעות טבלת δ (המתייחסת גם לאות שעבורה היה אי התאמה). חישוב טבלת δ לוקח $O(m^3|\Sigma|)$ אך לאחר מכן, חישוב באמצעות הטבלה לוקח O(n).

KMP
KMP – מחקה את האוטומט "ללא תלות" באות הקלט הבאה. סבוכיות בניית טבלת π – O(m). סבוכיות פרוצדורה ראשית O(n), סה"כ O(m+n). KMP יכול לייצר עוד פונקצית π(i) שערכה הוא גודל הרישא המקסימאלית של p שהיא סיפא של T[1…i] (ערך j בסיום ה-for).
(*) מספר הקריאות המקסימאלי ל-π הוא n-min(m-1,1) (n כאשר P מכיל תו בודד ו-T מורכב רק מתו זה, ו-m-1 עבור P גדול יותר).
i - מיקום בT, j - כמה התאמה יש. π[j] – לאן להזיז את P[1,…,j] אם הבדיקה נכשלה. **האלגו':** קוראים את האות הבאה של T ⟸ T[i]:
(1) אם P[j+1]=T[i] ⟸ נגדיל את j ב-1. (2) אם אין התאמה ⟸ נשנה את j ל- π[j]. (3) אם j=m=אורך P ⟸ נודיע שנמצאה התאמה ונשנה את j ל- π[j]. (נחזור על הבדיקה עד שתמצא התאמה או ש- j=0, ואז נקרא את האות הבאה בT ⟸ T[i+1]).
π(i) – בהינתן אינדקס i מוחזר לנו מה גודל הרישא ממש של P[1..i] שהוא גם סיפא ממש של P[1..i]. π מחושבת רק מתוך P וזמן חישובה הוא O(m). לדוגמא אם P=ABABAC π(4) יהיה 2, ואם התאמנו 4 תווים נזוז 2 מקומות. (גודל התזוזה שלנו הוא i-π(i)).
ניתן לחשב בזמן הרצת KMP גם את הפונקציה הבאה: **(i)ד** – בהינתן אינדקס i מוחזר לנו מה גודל הרישא המקסימלית של P שהיא סיפא של T[1..i]. בהרצת KMP זה פשוט ערכו של j בסיום לולאת ה FOR. מבחינה רעיונית זה שקול לפונקציה הבאה רק שהיא מקבלת מחרוזת ולא אינדקס: **σ(x)–** בהינתן מחרוזת x מוחזר לנו מה גודל הרישא המקסימאלית של P שהיא סיפא של x.
<u>חסמים הדוקים</u> – מס' השוואות מול תו אחד = m, מס' השוואות מול כל T = 2n-min(m-1,1).

9 הערות כלליות

(*) גרף לא מכוון ללא מעגלים שליליים מכיל רק משקולות חיוביות על הקשתות. (*) גרף מכוון בו מחפשים מסלולים באורך <u>מדויק</u> של x קשתות ניתן לבנות בצורה א-ציקלית. (*) גרף <u>לא מכוון</u> הוא **דו"צ** אמ"מ אין בו מעגל באורך אי זוגי. (*) אם עושים פיצול קשתות, אז כדי לשמור על יחידות נכפל את הקודקודים. (*) מספר זוגי של קשתות/אורך קשתות במסלול- בד"כ נשתמש בהכפלה של הגרף ושימוש באלגוריתמים של מק"בים. דרכים להתמודדות

עם **בעיות פרישה:** (1) שינוי המיון של קרוסקל. (2) הגדרת גרף המכיל רק קשתות עם משקל קטן/גדול מרף מסויים (ובדיקת קשירות בגרף זה בין רכיבים/צמתים). (3) הגדרת פונקצית משקל חדשה. **בעיות דו"צ:** (1) שימוש בצבעים. **קשירות:** (1) הסתכלות על גרף העל של הרק"חים. (2) שימוש ב-DFS לאיתור מעגלים (לשים לב לקשתות קדמיות). **מק"בים:** (1) מציאת מסלול קצר ביותר מכל צומת ב-v אל קבוצת צמתים "מיוחדים" – הוספת צומת חדש s וחיבור בקשתות עם משקל 0 לצמתים המיוחדים. חיפוש מסלולים קצרים ביותר מ-s לשאר הצמתים. (2) תרגיל תחנות הדלק עם הגבלת אורך מסלול – בדיקה אם במקורי קיים מסלול תחת ההגבלה. אם לא, מציאת מסלול קצר ביותר בין הצמתים "המיוחדים" לצמתים ה"מיוחדים האחרים ללא קשתות החורגות מהמגבלה ושמירה בגרף חדש מייצג (ללא קשתות כבדות) חיפוש מסלול קצר ביותר בגרף החדש ושחזור המסלול לצמתים מקוריים. (3) מציאת מסלול קצר ביותר באורך קבוע מראש – שכפול הגרף מספר דרוש של פעמים. בגרף המשוכפל אין מעגלים. **תתי סדרות:** כאשר יש תת סדרה שהיא לא במרחקים שווים אלא ב"לפחות" (ולא לכל היותר) נשתמש בגרידי. **כללי:** אם רשום א-ציקלי – ישר לחשוב על רק"חים.

10 תכנות דינאמי

שאלה: עבור **סדרת מספרים** ממשיים x1,x2,…,xn לחשב עבור כל i את מספר תתי הסדרות העולות ממש ומסתיימות ב- xj (כולל).
פתרון: ניצור מערך בגודל n מאותחל ל- 1, כי כל סדרה יכולה לכלול רק את האיבר האחרון. אח"כ נסכום עבור כל xi (כש- i<j) שקטן מה- xj את תתי הסדרות שלו, כי פשוט ניתן להוסיף בסוף את xj:
סיבוכיות: n^2.

```
List a[1…n] ← [1…1]
  For j= 1 to n
    For i= 1 to j-1
            If xi<xj then
              a[i]= a[j]+a[i];
```

שאלה: נתונות **2 סדרות** S ו-T. תארו אלגוריתם יעיל ככל האפשר שקובע אם S מופיע **כתת סדרה** בT כך שכל שני איברים עוקבים של S מופיעים במרחק של לכל היותר 50 זה מזה. **פתרון:** נכין טבלה M כך שM(i,j) יהיה 1 אם j האיברים הראשונים של S עונים על התנאים עבור i האיברים הראשונים של T (ו-j נופל על האיבר ה-i של T). כדי לחשב M(i,j) נבדוק האם T[i]=S[j] וגם [האם מתקיים M(i-1,j-1) או M(i-2,j-2) או ... M(i-50,j-1)]. **סבוכיות:** מילוי הטבלה יתקיים ב-O(mn).

שאלה: נתונה קבוצה של **n מספרים שלמים $1\le a_1<a_2\ldots<a_n<2n$** ומספר שלם נוסף m. תארו אלגוריתם יעיל המחשב את מס' הפתרונות **למשוואה $a_1x_1+a_2x_2+\ldots+a_nx_n=m$ כאשר $x_i\in\{0,1,2\}$.**
פתרון: נגדיר מטריצה עם עמודות $0\ldots m(\le 4n^2)$ ושורות עם a_i. בכל תא בטבלה נמלא את מספר הפתרונות הקיימים כאשר עבור m=לערך העמודה וכאשר הסדרה מכילה את $a_1..a_i$ בלבד. העמודה הראשונה תכיל 1 (כי יש פיתרון יחיד כשm=0). בשורה הראשונה נסמן 1 בתאים שווים ל-$0*a_1$, $1*a_1$ ו-$2*a_1$ ובשאר 0. נמלא את המטריצה לפי שורות ואז עמודות (ע"פ הקוד). **סבוכיות:** O(mn)

```
If 2ai≤j then
a[i,j]=a[i-1,j-2ai]+ a[i-1,j-ai]+ a[i-1,j]
else if ai≤j then
a[i,j]=a[i-1,j-ai]+ a[i-1,j]
else
a[i,j]=a[i-1,j]
```

שאלה: A מערך של n מספרים ממשיים. צ"ל אלגו' למציאת **תת קבוצה** של A **שסכומה מקסימאלי** אבל **אסור לבחור 2 איברים סמוכים מA. פתרון:** מטריצה M תכיל את הסכום המקסימאלי עם תתי סדרות המכילות רק איברים הקטנים שווים לi. מטריצה P תכיל זוג סדור עם קבוצה להוסיף לתת ופוינטר לאינדקס הבא של P. מאתחלים 2 תאים ראשונים. עוברים על המערך לפי התנאים שבקוד. לסיום, עוברים על המערך שוב מהסוף להתחלה ע"פ הפוינטר בזוג הסדור כשבכל שלב מוסיפים את האיבר שבחלק הראשון של הזוג הסדור.

```
if(A [i-2]+A[i]>A[i-1])
  M[i]←A[i-2]+A[i]; P[i]←<{i},i-2>
  else M[i]←A[i-1]; P[i]←<Ø,i-1>
```

שאלה: נתונה סדרת טבעיים בגודל n. תארו אלגו' שמוצא תת סדרה (לאו דווקא רצופה) מונוטונית לא יורדת עם סכום מקסימאלי. **פתרון:** מאתחלים את הראשון. אחרי זה רצים בלולאה לפי הקוד. בסוף עוברים על המערך ומוצאים את המיקום עם הסכום המקסימאלי. ממנו רצים באמצעות המצביעים ששמרנו ומשחזרים את תת הסדרה. **סבוכיות:** n^2

```
j=i-1; while j>0 and A[j]>A[i] j=j-1;
If j=0 then M[i]=ai; P[i]=Nil;
  else M[i]=M[j]+ai; P[i]=j;
```

שאלה: אותו דבר אבל **אורך תת הסדרה הוא לכל היותר k**. **פתרון:** מריצים אלגוריתם דומה, אבל לכל סכום סדרה שומרים גם כמה איברים יש עד כה ועותק ממש של כל איברי תת הסדרה. אם הגענו לתת סדרה הגדולה מk, נסיר מהעותק שלנו את האיבר הראשון (הוא הכי קטן) ונשים במקומו את האיבר הנוכחי (צריך גם לעדכן את הסכום להיות הסכום הקודם פחות האיבר הראשון + האיבר הנוכחי). **סבוכיות:** kn^2

שאלה: נתונה **סדרת מספרים ממשיים**, רוצים **תת סדרה רציפה** כך ש**הסכום מקסימאלי**. **פתרון:** מגדירים Si+ ו-Si- פתרון אופטימלי עם/בלי x_i. חישוב הערכים: $Si+=\max\{Si-1+x_i, x_i\}$, $S_i-=\max\{S_{i-1}+,S_{i-1}-\}$. בסוף $S=\max\{S_n+,S_n-\}$. סוף הסדרה= התא הראשון מהסוף כך ש-$S_i+=S$.

כפל מטריצות יעיל רוצים למצוא סדר הכפלה עם מספר מינימלי של פעולות. כפל בודד של מטריצה $p\times q$ במטריצה $q\times r$ הוא pqr.
$C[i,j]=$ המחיר המינימלי של כפל המטריצות $A_iA_{i+1}\cdot\ldots\cdot A_j$.

```
for i=1 to n do C[i,i]←0   //הפרש 0
for l=1 to n-1 do          //הפרש i
  for i=1 to n-l do    j←i+i;  C[i,j]← ∞;
                 for k=l to j-1 do   t←C[i,k]+C[k,j]+p_{i−1}p_k p_j
                                  if t<C[i,j] then C[i,j]←t; Π[i,j]←k
```

טריאנגולציה עם משקל מינימלי: נתון מצולע קמור p עם n קודקודים במישור. ע"י העברת מיתרים שלא חוצים זה את זה נפרק את p למשולשים. לכל מצולע יש $\binom{n}{3}$ משולשים פוטנציאליים. מטרה: לבנות טריאנג' עם משקל מינימלי. נפצל כל מצולע במיתר שממנו קבלנו אותו, וזה יבטיח שכל תת מצולע, לאורך כל הרקורסיה יהיה מורכב מחלק רציף משפת p + מיתר. נתאר את $P`$ ע"י (p_i,p_j) והכוונה לזה ששפתו היא מ-p_i עד p_j נגד כיוון השעון. יש בסה"כ $O(n^2)$ תתי מצולעים. $j=i+1$ מצולע ריק שהוא רק צלע. במקרה זה נאמר שמשקל המצולע הוא 0. $j=i+2$ המצולע משולש. $=W_{1n}$
$W_{ij}=min_{2\le k\le n-1}[w(p_1p_kp_j)+W_{1k}+min_{2\le k\le n-1}[w(p_1p_kp_n)+W_{1k}+W_{kn}]$
$W_{kj}]$. נפתור בעיות עם הפרש 1: $i=1,\ldots,n-1$ $W_{i,i+1}\leftarrow 0$. נרוץ בלולאה על ההפרש (j): $W_{i,i+j}=min[\ldots W_{i,k},W_{k,i+j}]$. **סבוכיות:** $O(n^3)$.

חישוב תת סדרה משותפת מקסימלית

```
For j=0 to n { LCS[0,j]← Ø }
For i=1 to m { LCS[i,0] ← Ø
           for j=1 to n do
             If Xi = Yj then
               LCS[i,j] ←LCS[i-1,j-1]||(Xi)
             Else
               LCS[i,j] ←max{LCS[i,j-1],
                             LCS[i-1,j]} }
```

11 תרגילים ופתרונות

מציאת רשת שיורית (residual network): בהינתן רשת N=((V,E),c,s,t), וזרימה ברשת f, הרשת השיורית היא $N_f=((V,E_f),c_f,s,t)$, כאשר פונקצית הקיבולות השיורית c_f מוגדרת על ידי $c_f(u,v)=c(u,v)-f(u,v)$ לכל u,v∈V, וקבוצת הצלעות השיורית E_f מוגדרת ע"י $E_f=\{(u,v):c_f(u,v)>0\}$. (שימו לב כי הרשת השיורית היא רשת זרימה).
בעיית מספר המסלולים הזרים: __קלט__: גרף מכוון G=(V,E) ושני קדקודים s,t∈V. __צ"ל__: מה **המספר המקסימאלי של מסלולים הזרים בצלעות** מ-*s* ל-*t*. __פתרון__: רדוקציה לבעיית זרימה מקסימאלית בשלמים. רשת הזרימה שנבנה היא N=(G,c,s,t), כאשר *c* היא פונקצית קיבול הנותנת לכל צלע קיבול בגודל 1. נפעיל על *N* אלגוריתם למציאת זרימה מקסימאלית. הזרימה המקסימאלית תהיה שווה בגדלה למספר המסלולים הזרים בצלעות. __שלבים עיקריים בהוכחה__: אם קיימת זרימה ב-*N* בגודל *k*, אזי קיימים ב-*G k* מסלולים זרים בצלעות מ-*s* ל-*t*. אם קיימים *k* מסלולים זרים בצלעות ב-*G*, אזי קיימת ב-*N* זרימה בגודל *k*
בעיות מרכזיות עם מעגלים שליליים:
__תרגיל__: נתונים n **סוגי מטבעות $c_1, \ldots, c_n$ וטבלת המרה** (מטריצה): $A_{nxn}=\{a_{i,j}\}$, כאשר a_{ij} מציין כמה מטבעות מסוג c_j נקבל תמורת מטבע אחד מסוג c_i (שער חליפין). רוצים לבדוק אם קיימת סידרת המרות שתביא לרווח, כלומר, אפשרות להתחיל ממטבע אחד מסוג כלשהו, נניח c_{i1}, להמיר ל-c_{i2}, אח"כ - $c_{ij}, \ldots, c_{ik}$ ואז חזרה ל-c_{i1}, כך שבסוף נישאר עם יותר ממטבע אחד מסוג c_{i1}. כלומר, רוצים לבדוק האם קיימים אינדקסים $i_1, \ldots, i_k$ כך ש: $a_{i1,i2}*a_{i2,i3}*\ldots*a_{k-1,k}*a_{ik,i1}>1$. __פתרון__: נוציא log משני האגפים ונקבל: $\log(a_{i1,i2})+\log(a_{i2,i3})+\ldots+\log(a_{k-1,k})+\log(a_{ik,i1})>0$, או: $-\log(a_{i1,i2})-\log(a_{i2,i3})-\ldots-\log(a_{k-1,k})-\log(a_{ik,i1})<0$. נבנה גרף מכוון שלם (כלומר, עם כל הקשתות האפשריות) שקודקודיו $c_1, \ldots, c_n$. לכל קשת (c_i, c_j) ניתן משקל $-\log(a_{ij})$. הבעיה שקולה למציאת מעגל שלילי בגרף. לשם כך נריץ את Bellman-Ford מצומת אקראי ונחפש מעגל שלילי הנגיש מצומת זה. כיוון שהגרף שלם, כל צומת נגיש מכל צומת ולכן נמצא מעגל שלילי אם קיים כזה. זמן ריצה לא משהו: $O(|V|*|E|)=O(n^3)$.

__טענה__: קיים **פתרון למערכת** ⟸⟹ אין מעגל שלילי בגרף G (המעודכן). __הוכחה__: (⟸) נניח שיש בגרף מעגל שלילי $C: x_{i1}, x_{i2}, x_{ik}, x_{i1}$. נסכם את השיוויונות המתאימים לפי הסדר ונקבל $0\le w(C)<0$ וזו סתירה.(⟹) נניח שאין בגרף מעגל שלילי, אזי Bellman-Ford מ-s מחזיר לכל x_i את $\delta(s, x_i)$, ומכיוון שכל הצמתים נגישים מ-s אזי $\delta(s, x_i)<\infty$ לכל i. נראה שאם נציב לכל x_i את הערך $\delta(s, x_i)$ נקבל פתרון למערכת. עלינו להראות שכל אי-שיוויון $x_j-x_i\le c_{ij}$ מתקיים, כלומר $\delta(s, x_j)-\delta(s, x_i)\le c_{ij}$. אבל $c_{ij}=w(x_j, x_i)$ ולכן מספיק להראות ש- $\delta(s, x_i)\le\delta(s, x_j)+w(x_j, x_i)$ ואת זה הוכחנו בכיתה. __מסקנה__: כל האי-שיוויונים מתקימים עבור הצבה זו.

__תרגיל__: היה לנו __גרף מכוון__ ורצינו לבדוק **אם קיים מעגל בגודל 0**. יצרנו קודקוד חדש שמחובר לכל הקודקודים עם קשתות במשקל 0, הרצנו BF ויצרנו את הגרף G' שהתקבל כש: $E'=\{(u, v)\in E|\ \delta(s, v)=\delta(s, u)+w(u, v)\}$. יש מעגל אם ב-G' קיימת קשת אחורית (בדיקה עם DFS).

__טענה__: קיים **מעגל שנגיש מ-s** ב-G' ⟸⟹ משקל מעגל ב-G הוא 0. (⟸) נניח שמעגל $C:v_1, v_2, \ldots, v_k, v_1$- נמצא ב-G' ונגיש מ-s. אזי: $\delta(s, v_2)=\delta(s, v_1)+w(v_1, v_2)$, $\delta(s, v_3)=\delta(s, v_2)+w(v_2, v_3)$,…, $\delta(s, v_1)=\delta(s, v_k)+w(v_k, v_1)$. נסכם את האגפים ונקבל $w(C)=0$.
(⟹) נניח $C:v_1, v_2, \ldots, v_k, v_1$ במשקל 0. נניח בשלילה שהוא אינו ב-G'. אזי קיימת קשת שלו שאינה ב-G'. נניח בה"כ שזו הקשת (v_k, v_1). אזי: $\delta(s, v_1)<\delta(s, v_1)+w(v_1, v_2)$, $\delta(s, v_2)\le\delta(s, v_2)+w(v_2, v_3)$, …, $\delta(s, v_k)\le\delta(s, v_{k-1})+w(v_{k-1}, v_k)$. סה"כ: $0<w(C)$ בסתירה לכך ש-$w(C)=0$.

קשת מנתקת (גשר), קודקוד מנתק, גרף דו קשיר, רכיב דו קשירות, גרף דו צדדי:
נחשב את **low** תוך כדי הרצת DFS על הגרף. החישוב יתבצע לכל קודקוד ברגע שמסיימים לטפל בו (לאחר חישוב ערך f שלו). זאת בעזרת הנוסחא:
low (v) = min {d[v], min{d[u]| (v, u) ∈E אחורית קשת}, min{low(w)| w של ישיר בן v}}.
__טענה:__ כל קודקוד v **שאינו שורש, v מנתק** ⟸⟹ קיים w בן של v כך ש: low(w) ≥d[v] __הוכחה:__ (⟹) אם קיימת w כנ"ל אזי אי-אפשר להגיע מ-w להורה של v ⟸ v מנתק. (⟸) אם v מנתק אזי קיים w כנ"ל, אחרת אחרי הסרת v, עדיין ניתן יהיה להגיע מכל בן של v (או צאצא שלו) לכל אב של v, כלומר הסרת v תשאיר את הגרף קשיר בסתירה להנחה ש-v מנתק. __טענה:__ אם-v __שורש__ אזי **v מנתק** ⟸⟹ יש לו לפחות 2 בנים. (מיידי).
מוצאים רכיבי דו-קשירות בדומה לקודקודים מנתקים. אלא שלכל u שאינו שורש בודקים את בניו. לכל בן v אשר low(v) גדול או שווה d(u) נפצל את u – עותק אחד עבור הקשת (u,v) ועותק אחד עבור השאר. עבור השורש נתאים עותק לכל בן.
__הגדרות:__ (1) בגרף __לא מכוון__, **קודקוד מנתק** הוא קודקוד שהסרתו מנתקת את רכיבי הקשירות שלו. (2) גרף __לא מכוון וקשיר__ הוא **דו-קשיר** אם אין לו קודקודים מנתקים. (3) **רכיב דו-קשירות** הוא **תת-גרף דו-קשיר מקסימלי**.
__טענה:__ לכל 2 רכיבי דו-קשירות יש לכל היותר קודקוד משותף אחד. __מסקנה:__ רכיבי הדו-קשירות מהווים חלוקה של קשתות הגרף. __הוכחה:__ כל קשת (u,v) בפני עצמה היא תת-גרף דו-קשיר, לכן היא משתתפת ברכיב דו-קשירות, ולפי הטענה האחרונה נובע שאין קשת המשותפת לשני רכיבים.
__הגדרה:__ **גשר** בגרף __לא מכוון__ היא קשת שהסרתה מנתקת את רכיבי הקשירות שלה. __טענה:__ קשת **אינה גשר** ⟸⟹ היא נמצאת על מעגל פשוט. __טענה:__ גרף __לא מכוון וקשיר__ ניתן **לכיוון לקשיר בחוזקה** ⟸⟹ אין בו גשרים. __טענה:__ גרף __לא מכוון__ הוא **דו"צ** ⟸⟹ אין **מעגלים באורך אי-זוגי**.

12 משפטים

MFMC: זרימה מקסימלית ⟸ הרשת שיורית לא מכילה מסילה מ s ל- t (אם הייתה קיימת היינו יכולים להגדיל את F).
רשת שיורית לא מכילה מסלול ⟸ קיים חתך שקיבולו שווה לזרימה (נבנה אותו S כל הצמתי שנגישים מ s ברשת השיורית ו T כל השאר. לכל u ב S ולכל v ב T f(u,v)=c(u,v) (אם לא אז יש ברשת השיורית מסלול מ s ל t בסתירה להנחה)).
קיבול של חתך שווה לזרימה ⟸ היא מקסימלית (יודעים שזרימה קטנה מקיבול כל חתך ולכן אם היא שווה לקיבול של אחד מהם אז היא מקסימלית).
ED: לכל צומת המרחק ברשת השיורית מ המקור עולה מונוטונית חלש (מניחים בשלילה שלא ונראה שלא יתכן) ע"י כך נוכיח שמספר ההזרמות בקשת קטן מ |V||E| .
Dinic: מספר הפאזות ב-dinic קטן מ |V|. נראה כי - $\delta_{f'}(s,t)>\delta_f(s,t)$. כל קשת (u,v) ברשת השיורית G_f, או שהיתה קשת של G_f ואז מתקיים: $\delta_f(s,v)\le\delta_f(s,u)+1$. אחרת – (v,u) הייתה קשת ואז מתקיים: $\delta_f(s,u)=\delta_f(s,v)+1$ כלומר- $\delta_f(s,v)=\delta_f(s,u)-1<\delta_f(s,u)+1$. מכאן שאם $s=v_o\ldots v_k=t$ מק"ב ב-$G_{f'}$, אז $\delta_f(s,v_i)\le\delta_f(s,v_{i+1})+1$ ובאינדוקציה על i: $\delta_f(s,v_i)\le i$, כלומר- $\delta_f(s,t)\le k=\delta_{f'}(s,t)$. שיווין מתקיים אמ"מ כל קשת (v_i,v_{i+1}) הייתה ב-G_f, אך זה לא אפשרי משום שזו מסילה מ-s ל-t ב-G_f, בסתירה לכך ש-f' התקבלה מ-f ע"י תוספת של זרימה חוסמת. לכן קיבלנו אי-שיווין חזק.
טיפוס 2: מציאת זרימה חוסמת ברשת מיוחדת זמן O(E).(כי מסילה שמוצאים זורקים את כל קשתותיה). מספר הפאזותת ברשת זו הוא $O(\min(|E|^{\frac{1}{2}},|V|^{\frac{2}{3}}))$. (לאחר $E^{1/2}$ פאזות הזרימה גדולה מ $E^{1/2}$. כעת יש ברשת השיורית חתך בגודל לכל היותר $E^{1/2}$ ולכן זו הזרימה המקסימלית בו היא $E^{1/2}$ לכן אם *f היא ערך זרימת המקסימום אז f*-f הוא לכל היות $E^{1/2}$. לכן עוד לכל היותר $E^{1/2}$ פאזות נסיים. אותו טריק להראות את $|V|^{2/3}$). נכון גם אם הייתה רשת 0/1/2.
טיפוס 1: מספר הפאזות קטן מ $V^{1/2}$. לאחר $V^{1/2}$ פאזות המרחק בהכרח גדול מממספר זה. קיים $|Vi|<|V|^{1/2}$/|V| ולכל צומת אם נסלק את הקשת היחידה שנכנסת אליו או יוצאת ממנו זה יהיה חתך ולכן מצאנו חתך ונמשיך כמקודם. נכון גם אם הייתה רשת 0/1/2.

13 אלגוריתמים הסתברותיים

קיום זיווג מושלם עם בדיוק m קשתות כחולות: נגדיר מטריצה $A_{xy}=\{0,(x,y)\notin E;\ wZ_{xy},(x,y)\in E(blue);\ Z_{xy},(x,y)\in E\ \}$ (נגדיר משתנה Z אם קיימת קשת מ-x ל-y וניתן לה מקדם w אם היא כחולה, אחרת – אפס). קיים זיווג מושלם כמבוקש ⇔ w מופיע בפולינום הדטרמיננטה עם מקדם שונה מאפס ובחזקת m. לכן, נציב $k=2^nn^{3n}$ במקדם w ונגריל מספרים בין 1 ל-2n ונציב במשתנים Z, נחשב דטרמיננטה (ע"י דירוג) ונבטא אותו בבסיס k. אם עבור הצבה מסוימת יש מקדם שונה מאפס ל-k^m - בהכרח יש זיווג כנ"ל, אחרת – בסיכוי קטן מ-$\frac{1}{2}$ אין זיווג מושלם כנ"ל (וניתן לחזור על האלגוריתם כדי להקטין את הסיכוי כרצוננו).
בדיקת k-קשירות: קב' שולטת בגרף היא $U\subseteq V$ כך שלכל $v\in V$ או ש-$v\in U$ או של-v שכן ב-U. נמצא קבוצה שולטת קטנה בתוחלת $E(|U|)=O(\log n)$, ע"י הגרלת קבוצה X שצומת שייך אליה בסיכוי כלשהו p. נגדיר קבוצה שולטת $U=X\cup(V\backslash X\backslash N(X))$ (העונה על הדרישות). נבדוק שיש k מסלולים זרים בקשתות בין כל זוג צמתים בקבוצה השולטת U (בדרך הרגילה - זרימה). נקבל __סיבוכיות:__ $O(\log^2 n|E|\min(|E|^{\frac{1}{2}},|V|^{\frac{2}{3}},k))$.
עץ החלטות (NAND): נרצה לחשב את ערך הפונקציה NAND על הקלט $x_1..x_n$ עם מעבר על כמה שפחות מהקלט. (טענה: כל עץ החלטה דטרמיניסטי המחשב NAND צריך לקרוא את כל הקלט). לכן, נבנה עץ החלטה ונפעיל את האלגוריתם הבא: נבחר בן מקרי ונחשב רקורסיבית את ערכו. אם קיבלנו 0, נחזיר 1, אחרת נאלץ לחשב גם את הבן השני ונחזיר את השלילה שלו.
$T_0(d)$ תוחלת מס' הביטים שיש לקרוא בעץ מדרגה d כשהתוצאה היא 0. $T_1(d)$ באופן דומה עבור תוצאה 1. נחסום באופן רקורסיבי:
$T_0(d)\le T_0(d-1)+\frac{1}{2}T_1(d-1)$, $T_0(d)\le 2T_1(d-1)$.