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.

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).
(כדי לחלק את הגרף לקבוצות, נגדיר v1 להיות כל הצמתים בשכבות הזוגיות ו-v2 בשכבות האי זוגיות)

3 DFS

שימושים: (1) מציאת רכיב קשירות - 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 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 טפולוגי 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 בגרף מכוון.**
(*) 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).
טענה: לכל 2 רכיבי **דו קשירות** יכול להיות רק **קודקוד משותף** 1. **מסקנה:** רכיבי דו"ק מהווים חלוקה של הקשתות לקב' זרות.

4 עץ פורש מינימאלי (עפ"מ)

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|V|)=O(|E|log|E|).
(*) הפלט של האלגוריתם של קרוסקל תלוי אך ורק בסדר הממוין של הקשתות.
(*) אם לכל הקשתות יש משקלים שונים אז אלגוריתם קרוסקל יוצר עץ אחד ויחיד.
(*) האלגוריתם של קרוסקל מסוגל לייצר כל עפ"מ אפשרי. **מסקנה:** אם המשקלים שונים, יש עפ"מ יחיד.

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 $f:\mathcal{R}\rightarrow\mathcal{R}$ פונקציה עולה ממ, אם נגדיר משקל חדש w*(e)=f(w(e)) ז T 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|+|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.

6 All-Pairs Shortest Paths

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

FW
פורד וורסל – 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
Johanson - 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 $h:V\text{-}\mathbb{R}$ 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 פלקינסון
פורד פלקינסון – 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 המקסימאלית 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)).

אדמונד קארפ
אדמונד 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 $O(|E|)$ 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 המקסימליות (נגזר ממספר השכבות המקסימאלי) – O(|V|). ראה/י גם רשתות 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|) (מדויק יותר-$O(|E||V|^2)$). לעיתים ניתן למצוא חסם הדוק יותר ע"פ טיפוס.
<u>רשת מטיפוס 1</u> – רשת בה אין קשתות מקבילות (או אנטי מקבילות). **סבוכיות:** $O(|E||V|^{2/3})$. <u>רשת מטיפוס 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!
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.

זיווג מושלם
משפט Hall – יהיה G גרף דו"צ $V = X \overset{\cup}{_{\text{זר}}} Y$, 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 $A\subseteq X$ 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.

8 תכנות דינאמי

בקרוב כמה דוגמאות פה.

9 התאמת מחרוזות

בשיטה נאיבית נוכל לבצע השוואה מכל מקום במחרוזת T. זה מבוצע בזמן O(mn). ראינו שיטת השוואות באמצעות טבלת π 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 τ(i) שערכה הוא גודל הרישא המקסימאלית של p שהיא סיפא של T[1…i] (ערך j בסיום for).

10 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.