תרגול מס' 10 – אלגוריתם דיניץ

הגדרה:

רשת שכבות : תהי N_f רשת שיורית אשר קיים בה מסלול קצר ביותר מ-s אל t באורך k.
שכבה i של רשת השכבות Λ_f עבור $0 \le i \le k$ מכילה את הקודקודים $V_i = \{u \in V \mid d_f(u) = i\}$,
כאשר $d_f(u)$ הוא המרחק המינימאלי בצלעות מ- s ל- u ב (V, E_f).
E_f מכילה את כל הצלעות בין השכבות. $E_i = \{(u,v) \in E_f \mid u \in V_{i-1}, v \in V_i\}$.
גרף השכבות השיורי יהיה $L_f = (V_0 \cup V_1 \cup .... \cup V_k, E_1 \cup .... \cup E_k)$.
רשת השכבות תסומן: $\mathsf{L}_f = (L_f, c_f, s, t)$.

תיאור סכמטי של האלגוריתם :

א. אתחול הרשת השיורית N_f, וזרימה $f \leftarrow 0$.
ב. כל עוד יש מסלול מ-s ל-t ברשת השיורית בצע:
   a. בנה רשת שכבות L_f עבור N_f.
   b. מצא זרימה חוסמת b לרשת השכבות L_f.
   c. עדכן $f \leftarrow f + b$, ועדכן את N_f.

תיאור בניית רשת שכבות: השכבה V_i מכילה את הקדקודים שמרחקם המינימלי מ-s הוא i.

א. הכנס את s ל- V_0, $i \leftarrow 0$.
ב. כל עוד V_i אינו ריק ו-t לא שייך ל- V_i בצע:
   a. עבור כל קדקוד u ב- V_i בצע:
      עבור כל שכן v של u, אם v לא נמצא באף V_j, $j \le i$, הוסף את v ל- V_{i+1}
      והוסף את (u,v) ל- L_f.
   b. $i \leftarrow i + 1$
ג. אם t לא שייך ל V_i, אין מסלול בין s ל t.
   אחרת (t שייך ל V_i) : הפלט הוא רשת השכבות.

תיאור אלטרנטיבי לבניית רשת שכבות:

א. הרץ BFS מקודקוד s, ולכל קודקוד v, יהי $d(s,v)$ המרחק מ-s ל-v כפי שנמצא ע"י הרצת ה-BFS.
ב. אם $d(s,t) = \infty$: אין מסלול בין s ל t.
ג. אחרת:
   a. לכל $(u,v) \in E$ בצע:
      i. אם $d(s,u) + 1 = d(s,v)$ הוסף את (u,v) ל-L_f.
   b. החזר את L_f.

Algorithm 1 Dinitz *(G, c, s, t)*

$f \leftarrow 0$; construct the residual network $N_f = (G_f, c_f, s, t)$;

while there is a path from *s* to *t* in G_f **do**

≤ *invariant assertion: f is a flow in N ;*

construct the layered network $L_f = (L_f, c_f, s, t)$;

find a blocking flow *b* for L_f ; ≤ *assertion :* $d_{f+b}(t) > d_f(t)$;

$f \leftarrow f + b$;

construct the residual network N_f ;

≤ *Post-condition of while loop : f is a maximum flow in N* ;

Algorithm 2 LayeredNetwork_Construction *(N_f)*

$V_0 \leftarrow \{s\}$; $i \leftarrow 0$;

while $(V_i \neq \phi)$ and $(t \notin V_i)$ **do**

$V_{i+1} \leftarrow \phi$ **;** $E_{i+1} \leftarrow \phi$ **;**

for each $u \in V_i$ **do**

for each $v \in V$ such that $(\langle u, v \rangle \in E_f)$ and $(v \notin V_j, \forall j \leq i)$ **do**

if $(v \notin V_{i+1})$ **then**

add *v* to V_{i+1} ;

add $\langle u, v \rangle$ to E_{i+1} ;

$i \leftarrow i + 1$;

if $(V_i = \phi)$ **then**

return $L_f = (\phi, c_f, s, t)$; ≤ *assertion: there is no path from s to t in* N_f

$L_f \leftarrow (V_0 \cup \ldots \cup V_i, E_1 \cup \ldots \cup E_i)$;

return $L_f = (L_f, c_f, s, t)$;

Post-condition : if t is reachable from s in N_f *, then* L_f *is the layered network for* N_f

Algorithm 3 BlockingFlow *(£$_f$)*

1. $b \leftarrow 0$; $M = (V_M, E_M) \leftarrow L_f$; $c \leftarrow c_f$;
2. **repeat**
3. ≤ *assertion : s is the only vertex in M with zero indegree ;*
4. find a path p from s to t in M; let its bottleneck capacity be $c_M(p)$;
5. **for each** edge $\langle u,v \rangle \in p$ **do**
6. $b(u,v) \leftarrow b(u,v) + c_M(p)$; $b(v,u) \leftarrow -b(u,v)$;
7. $c_M(u,v) \leftarrow c_M(u,v) - c_M(p)$;
8. **if** $c_M(u,v) = 0$ **then**
9. remove $\langle u,v \rangle$ from M;
10. ≤ *cleanup of M*;
11. if *indegree (v) = 0* then
12. CleanForward (v, M) ;
13. **until** *indegree (t) = 0* ;

Algorithm 4 CleanForward *(u, M)*

Pre-condition: indegree (u) = 0

for each v such that $\langle u,v \rangle \in M$ **do**
remove $\langle u,v \rangle$ from M;
if *indegree (v) = 0* **then**
CleanForward *(v, M);*

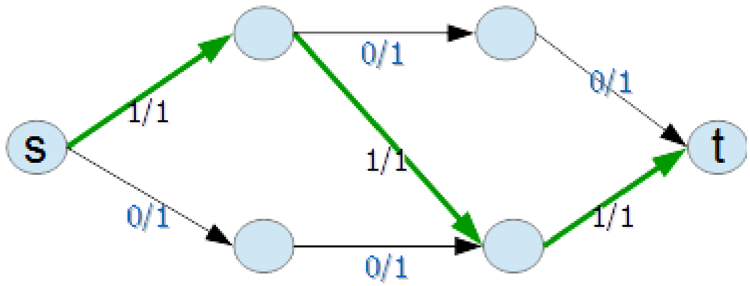
Post-condition: Let x be any vertex such that at the start of CleanForward *all paths back from x in M end at u. Then all edges going out of x have been removed.*

הערות חשובות על אלגוריתם דיניץ:

המושג "שלב" באלגוריתם יציין איטרציה בודדת של הלולאה באלגוריתם 1 (עדכון זרימה על סמך זרימה חוסמת ברשת השכבות).

שורה 4 באלגוריתם 3: מציאת מסלול ברשת השכבות מתבצעת על ידי סריקה אחורה מ-t עד ל-s. כיוון שדואגים לנקות את הרשת בכל שלב (CleanForward), מובטח כי סדרת הצעדים בהכרח תוביל בחזרה ל-s, כאשר כל קשת היא בין שכבות עוקבות. הזמן הדרוש למציאת מסלול ברשת השכבות הוא כסדר גודל מספר השכבות.

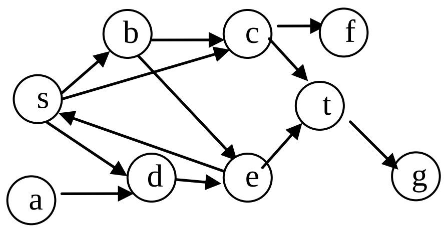
המושג "זרימה חוסמת" ברשת השכבות: שימו לב כי זרימה זו אינה בהכרח זרימה מקסימאלית ברשת השכבות. בשלב זה לא מעדכנים את קיבולי הקשתות ההפוכות לקשתות בהן עברה זרימה על ידי קשתות חרטה (אלו הן קשתות משכבה לשכבה אשר לפניה, ולכן הן לא חלק מרשת השכבות). בדוגמא הבאה ניתן לראות זרימה (בירוק) חוסמת ברשת השכבות שאיננה מקסימאלית בה:



שלא כמו באלגוריתם Ford-Fulkerson, אשר לא שומרים שום מידע מתהליך מציאת מסלול שיפור למציאת המסלול הבא בו "נעביר" זרימה, האלגוריתם של דיניץ מנצל את מבנה הנתונים "רשת השכבות" על מנת למצוא בעזרת סריקה אחת ב- G_f מספר מסלולים קצרים ביותר, מבחינת מספר קשתות.

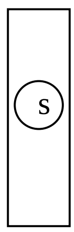
דוגמא לבניית רשת שכבות:

נתון הגרף הבא :



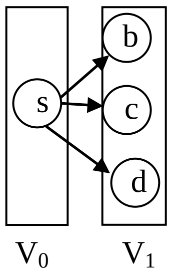
קודקוד המקור הוא s, קודקוד הבור הוא t.

שלבי בניית רשת השכבות יהיו כדלקמן :
בשלב הראשון, שכבה בודדת אחת:

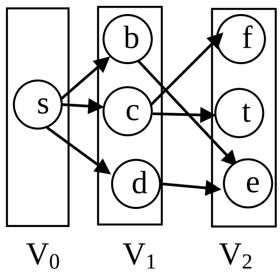


V_0

בשלב השני, שתי שכבות:



בשלב השלישי, שלוש שכבות:



כעת t מצוי בשכבה האחרונה, כלומר קיים מסלול מ-s ל-t ברשת השיורית, ואין לנו צורך בשכבות נוספות לעת עתה.

<u>ניתוח זמן ריצה עבור שלב אחד:</u>

1. אתחול – בניית הרשת השיורית: $O(|E|)$.
2. בניית רשת שכבות: (על ידי BFS מ-s עד t): $O(|E|)$.
3. נגדיר את זמן מציאת זרמה חוסמת לרשת השכבות ע"י $t(|E|,|V|)$.
4. עדכון הזרימה: $O(|E|)$.

סה"כ זמן ריצה עבור שלב: $O(|E|+t(|E|,|V|))$

ננסה כעת לספק חסם עבור מציאת הזרימה החוסמת (t (|E| + |V| :

<u>תיאור מציאת זרימה חוסמת לרשת שכבות</u> L_f:

א. אתחל זרימה חוסמת $b \leftarrow 0$.

ב. כל עוד יש מסלול P מ-s ל-t ב- L_f בצע:

a. עדכן את b, L_f לפי מסלול P (בדומה ל-FF ללא הוספת קשתות חרטה נגדיות ב-L).

b. עבור כל קשת (u,v) שהתאפסה בעקבות שינוי הזרימה במסלול P

i. מחק את (u,v) מ- L_f.

ii. אם דרגת הכניסה של v היא אפס וגם $v \neq t$. בצע ניקוי-קדימה (v, L_f).

ג. החזר b.

<u>תיאור ניקוי-קדימה $(v,(V,E))$:</u>

לכל קשת (v, x) (קשת היוצאת מ-v) בצע:

a. מחק את (v, x).

b. אם דרגת הכניסה של הקודקוד x היא 0 וגם $x \neq t$ בצע: ניקוי-קדימה$(x,(V,E))$.

<u>ניתוח זמן ריצה עבור מציאת זרימה חוסמת לרשת שכבות</u> $t(|E|,|V|)$:

1. מציאת מסלול ברשת השכבות $O(|V|)$ - כיוון שאנו מבצעים "ניקוי", לכל קודקוד שמגיעים אליו בסריקה הפוכה מ- t יש צלע נכנסת, ולכן סריקה זו מובילה ל-s (לכול קודקוד יש דרגת כניסה גדולה מ-0, מלבד אולי t עצמו). מציאת מסלול ברשת השכבות מתבצע $O(E)$ פעמים, כיוון שבכל פעם מוצאת קשת אחת, לפחות, מרשת השכבות. סה"כ הזמן הדרוש למציאת כל המסלולים אם כן הוא $O(|E|\cdot|V|)$.
2. עדכון L_f: $O(|V|)$, מתבצע $|E|$ פעמים לכל היותר, ולכן $O(|E|\cdot|V|)$.
3. כדי לחשב את הזמן הכללי של ניקוי רשת השכבות נשתמש בשיטת פיזור החיובים (ניתוח פחת):
   a. כל קשת יכולה להימחק לכל היותר פעם אחת $\Leftarrow O(|E|)$ עבור כל הקשתות.
   b. בדיקת הדרגות עבוד קודקודים מתבצעות בכל פעם שאנו מוחקים קשת (ניתן לבדוק דרגת קודקוד בזמן קבוע ע"י שמירת מוני דרגה) $\Leftarrow O(|E|)$ עבור כל בדיקות הקודקודים.

לפיכך קיבלנו את החסם: $t(|E|,|V|)=O(|E|\cdot|V|)$.

אם כן, זמן ריצה עבור שלב אחד באלגוריתם הוא $O(|E|+t(|E|,|V|))=O(|V|*|E|)$.
ישנם לכל היותר $|V|$ שלבים (בשיעור ראיתם כי בכל שלב מספר השכבות ברשת השכבות גדל, כאשר הוא חסום ע"י $|V|$), לכן זמן הריצה לאלגוריתם כולו הוא $O(|V|^2\cdot|E|)$.
ניתוח זמן הריצה עבור אלגוריתם דיניץ המפורט לעיל נכון עבור גרף כללי עם משקולות כלשהן. אך ישנם גרפים עבורם ניתן למצוא חסמים טובים יותר. <u>האלגוריתם עצמו לא ישתנה</u>, אך הניתוח יהיה עדין יותר ויתבסס על תכונות הגרף.

<u>דוגמה: גרפים בעלי קשתות בקיבול 1 בלבד:</u>

נראה שבגרפים כאלה ניתן להגיע לחסם זמן ריצה של $O(|E|\sqrt{|E|})$.
כדי להראות זאת, נראה שבגרפים בעלי קיבול 1 ניתן לחסום בצורה הדוקה יותר את עלותו של כל שלב ואת מספר השלבים.

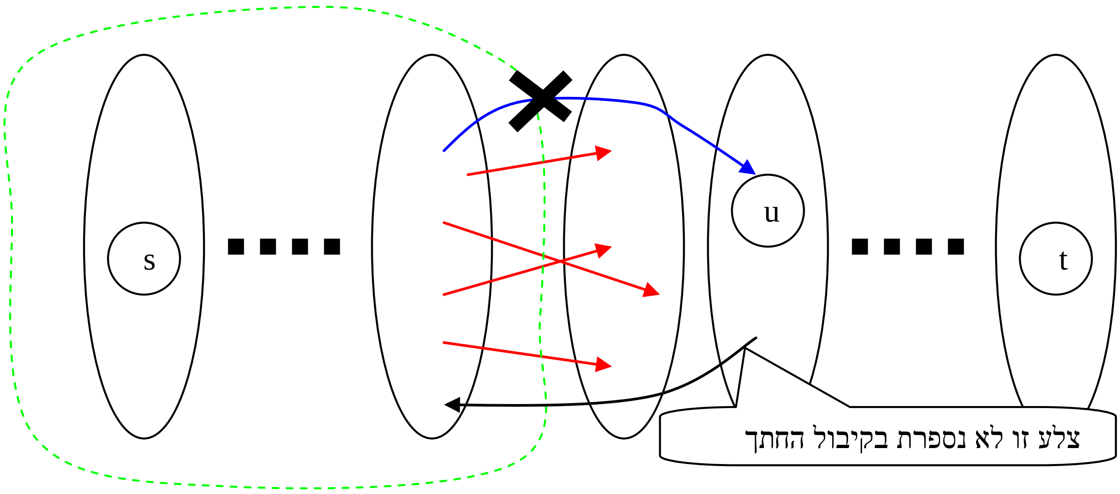
א. נראה קודם כי <u>עלותו של כל שלב</u> באלגוריתם אורכת $O\left(|E|\right)$ בלבד, במקום החסם : $t(|E|,|V|)=O(|E|\cdot|V|)$

הוכחת א:
נתבונן בגרף הרשת השיורית בעלת m שכבות, בתחילת שלב כלשהו. ברשת זו, אורך כל מסלול שיפור הוא $(m-1)$. כמה מסלולי שיפור יכול האלגוריתם למצוא?
בהינתן זרימה b ברשת השיורית נסתכל על סכומי הזרימה בצלעות שנמצאות ברשת השיורית בלבד. כמובן שערך זה גדול מ-|b|, מפני שאנו סופרים זרימה לאורך כל השכבות ולא רק הצלעות שיוצאות מ-s. נסמן את סכומי הזרימה בצלעות שנמצאות ברשת השיורית בלבד ב-$sum_m(b)$.
תחילה, $|b|=0$ ולכן $sum_m(b)=0$. בנוסף, $sum_m(b)$ תמיד חסום ע"י סך קיבולי הצלעות שנבחרו לרשת השכבות. לכל צלע ברשת השכבות הקיבול שווה ל-1 או 2, יש לכל היותר $|E|$ צלעות לכן תמיד יתקיים : $sum_m(b)\le 2|E|$.
כעת, בכל מסלול יש $(m-1)$ צלעות וצוואר הבקבוק הוא 1 או 2. לפיכך, לאחר בחירת מסלול שיפור $sum_m(b)$ גדל ב-$(m-1)$ או $2(m-1)$.
התוצאה היא שניתן למצוא לכל היותר $\frac{2|E|}{m-1}$ מסלולי שיפור.

כעת ניתן לראות שכל שלב עולה לכל היותר $O\left(|E|\right)$:

עדכון L ומציאת מסלול עולים $O(m)$ ומתבצעים לכל היותר $\dfrac{|E|}{m-1}$ פעמים (אלו הן הפעולות היחידות שעולות יותר מ-$|E|$ בניתוח הכללי). ניזכר שזמן בניית רשת השכבות הוא $O\left(|E|\right)$ וכך גם הזמן הכללי של פעולת הניקוי בה. כלומר, כעת גם החלק "הכבד" בשלב, אורך זמן $O\left(|E|\right)$. כעת ניתן לומר כי זמן הריצה של כל האלגוריתם חסום ע"י $O(|V|\cdot|E|)$.

ב. עתה נראה כי ניתן להקטין גם את החסם עבור מספר השלבים שמתבצעים ע"י האלגוריתם.
נגדיר בתחילת כל שלב חתכים ברשת השיורית $(S_i, V-S_i)$ כאשר S_i היא קבוצת כל הקודקודים במרחק עד i.



ברשת השיורית המצוירת לא ייתכן שיש קשתות משכבה i לשכבה גדולה מ-$i+1$. הדבר נובע באופן ברור מאופן בניית רשת השיורית על ידי האלגוריתם. (בדוגמא הקשת הכחולה (זו עם ה-X למדפיסים בשחור לבן) לא תיתכן ברשת השיורית הנ"ל, כלומר הקשתות האדומות הן היחידות שיוצאות מהחתך הירוק (השמאלי)).

מסקנה: החתכים המפרידים את קודקודי הגרף בין שכבה i לבין השכבה העוקבת זרים בצלעות. (ההפרדה נעשית בין שכבות עוקבות ברשת השיורית): ישנם $(m-1)$ חתכים כאלה אשר כולם זרים בקשתות, וכן ישנן לכל היותר |E| קשתות. לכן מספר הקשתות הממוצע לחתך הוא לכל היותר $\frac{|E|}{m-1}$. לפיכך, קיים חתך אשר מספר הקשתות בו הוא לכל היותר $\frac{|E|}{m-1}$. כיוון שקיבולי כל הקשתות הם לכל היותר 2 (מכיוון שבגרף המקורי כל הצלעות הם בקיבול 1), קיבולו של חתך זה חסום על ידי $\frac{2|E|}{m-1}$. מכאן נובע שקיים חתך ברשת השיורית שקיבולו הוא לכל היותר $\frac{2|E|}{m-1}$.

ניזכר בטענה: "הזרימה המקסימאלית שווה לחתך המינימאלי". על פיה, הזרימה המקסימאלית ברשת השיורית בשלב זה היא לכל היותר $\frac{2|E|}{m-1}$ וכל מסלול שיפור מגדיל את הזרימה ב-1 לכל הפחות (וב- 2 לכל היותר). לכן $\frac{2|E|}{m-1}$ הינו חסם על מספר מסלולי השיפור שעדיין ניתן למצוא ברשת השיורית!
נשים לב שאמנם ייתכנו עוד מסלולים נוספים ברשת השיורית, אשר אינם מופיעים בגרף הרשת השיורית. אך יחד עם זאת, גם מסלול כזה (מסלול לקודקוד t שאורכו גדול מ- m-1) ייאלץ לעבור ב- m-1 מבין הקשתות שכבר הושמו ברשת השכבות (קשת אחת לפחות המקשרת בין שכבה לעוקבת אחריה), ולפיכך הרוויית אותו חתך מינימאלי תחסום גם מסלול כזה.

טענה (ללא הוכחה):
אם ברשת השיוריות G_f הזרימה המקסימלית היא f' אזי הזרימה המקסימלית שווה ל-$f+f'$.
(נסו להוכיח זאת בבית כתרגיל נחמד לשעות הפנאי).

עוד נשים לב כי בכל שלב אנו מוצאים לפחות מסלול שיפור אחד (כל שלב מוצא את המסלולים הקצרים ביותר שקיימים בגרף). לכן יהיו לנו לכל היותר $\frac{2|E|}{m-1}$ שלבים עתידיים אחרי השלב הנוכחי. לפיכך, אם נבצע את החישוב עבור שלב כלשהו באלגוריתם בו מספר השכבות בגרף השכבות הוא m, הרי שבסך הכול היו לנו לכל היותר m שלבים ראשונים (עד לנקודה שבה המסלול הקצר ביותר הוא m), ועוד לכל היותר $\frac{2|E|}{m-1}$ שלבים להרוויית שאר הזרימה.

דבר זה נובע מכך שבכל שלב משפרים את הזרימה לפחות ב 1, אך לפי האבחנה לא ניתן לשפר ביותר מגודל זרימה מקסימלית ברשת השיורית,ולפי מה שראינו זרימה מקסימלית ברשת השיורית היא לכל היותר $\frac{2|E|}{m-1}$ (החתך שמצאנו). לכן יש עוד לכל היותר $\frac{2|E|}{m-1}$ שלבים.

סה"כ קיבלנו ביטוי למספר השלבים בהרצת האלגוריתם: $m+\frac{2|E|}{m-1}$.

מכיוון שהביטוי $m+\frac{2|E|}{m-1}$ נכון עבור כל $|V|>m>1$ אזי נחפש את החסם העליון ההדוק ביותר לביטוי (אנו רוצים למצוא את המספר המינימלי של האיטרציות).

מתי ביטוי זה יקבל את הערך המינימאלי? נגזור הביטוי לפי m, נשווה לאפס ונקבל את המינימום בשלב בו $m=\sqrt{2|E|}$. לכן לאורך כל האלגוריתם יהיו לכל היותר $\sqrt{2|E|}+\frac{2|E|}{\sqrt{2|E|}}=O(\sqrt{|E|})$ שלבים.

הוכחנו כבר כי כל שלב אורך $O(|E|)$. אי לכך, קיבלנו כי סך כל זמן הריצה הינו $O(|E|\sqrt{|E|})$.

<u>הסבר נוסף למה $m=\sqrt{E}$ (מספר האיטרציה באלגוריתם):</u>

בה"כ ניתן להניח ש- $m=\sqrt{E}$ אך אם אין כזאת, נסתכל על שתי איטרציות עוקבות, כאשר בראשונה יש פחות מהכמות הנדרשת של שכבות ובאיטרציה העוקבת יש יותר מהכמות הזו.
נסתכל על השלב ה –m+1 ועל הרשת השיורית הנוצרת בשלב הזה.
אזי בהכרח יהיו לנו בגרף של הרשת השיורית (אותו הגרף כמו שהוגדר ב-ב') לפחות m+1 שכבות.
מכיוון שבס"כ יש לנו $|E|$ צלעות אזי משובך היונים המוכלל: קיים לנו חתך בין שכבות בגודל קטן מ- $\sqrt{E}$. אחרת, מכיוון שהצלעות בין החתכים שונות אזי כל חתך גדול מ- $\sqrt{E}$ ונקבל:
$\sqrt{E}*(m+1)>\sqrt{E}*m=E$ שזה לא יתכן.
לכן נקבל שיש לנו חתך בגודל $\sqrt{E}$ לכן זה יקח לנו $\sqrt{E}$ איטרציות לכל היותר.

<u>שובך היונים המוכלל</u>

אם יש m תאים בשובך שלתוכם יש להכניס n יונים, אז בהכרח בתא אחד יהיו לפחות $p=\frac{n}{m}$ יונים או יותר(p הוא המספר השלם הקרוב (בעיגול כלפי מעלה) למספר $\frac{n}{m}$) :כלומר יש תא שמספר היונים בו הוא לפחות כמו הממוצע.