מס. מחברת 59

מבחן באלגוריתמים

5 - 1
10 - 2
9.5 - 3
2 - 4
10 - 5
10 - 6

סמסטר א' תשס"ט, מועד א'

תאריך: 6.3.09

מרצה: מיכה שריר

מתרגל : דן פלדמן

משך הבחינה: 3 שעות.

חומר עזר מותר: דף A4 אחד, כתוב משני הצדדים.

במבחן 6 שאלות. יש לענות על כולן.

- תשובות נכונות ומלאות על 5 מהשאלות יזכו אותך ב- 90 נקודות, ותשובות נכונות על כל השאלות ב- 100 נקודות.
- התשובה לכל שאלה מורכבת בד"כ משני חלקים, שעל כל אחד מהם להופיע במסגרת המתאימה. יש להשתדל לקצר בהסברים, ולא לחרוג מן המסגרות שהוקצו להם.
- מחברת הבחינה משמשת כטיוטא בלבד, אך יש למסרה.
- ודאו היטב את תשובתכם לפני כתיבתה בטופס המבחן. בסוף הטופס מצורף זוג מסגרות נוסף, לשימוש במקרי "חירום".
- התשובה לכל שאלה העוסקת באלגוריתם צריכה להיות יעילה ככל האפשר, ומלווה בהסבר מתאים.
- בכל השאלות שמתייחסות לגרפים, אם לא מצוין אחרת אז הכוונה לגרף פשוט (בלי לולאות ובלי קשתות מקבילות).

בהצלחה!

מס. מחברת 59

שאלה 1

נתון גרף מכוון G=(V, E), המיוצג ע"י רשימות שכנות, ונתונה תת-קבוצה E'⊆E של קשתות "חשובות".

תארו אלגוריתם יעיל ככל האפשר, הבודק אם קיים ב- G מסלול העובר דרך כל הקשתות של E' .

יעילות: $O(|E|\cdot|V|)$

מאוד לא יעיל

5

אלגוריתם:

~~נבנה את הגרף המושרה ע"י G': G-ב E'... נבדוק האם קיים ב-G מסלול אוילר. ואם כן, נחזיר "כן" ואחרת נחזיר "לא".~~

ניתן לכל הקשתות משקלים באופן הבא: לקשת חשובה ניתן משקל 1- ולקשת לא חשובה ניתן משקל 0.

נוסיף ל-V קודקוד חדש s ונחבר אותו לכל $u\in V$ בקשת מכוונת (s,u) במשקל 0.

נמצא את כל המקבילים מ-s לכל קודקוד באמצעות ע"י BF (אלא אם יש מעגלים שליליים).

נעבור לאחר B-F על כל ערכי d[u] לכל u. אם מצאנו u המקיים $d[u]=-|E'|$ נחזיר "כן", אחרת נחזיר "לא".

הסבר:

~~נניח שיש מסלול כזה ... G' הגרף המושרה ... מעגל אוילר ...~~

נניח קיים מסלול כנדרש העובר בכל קשתות E', המתחיל בקודקוד u ונגמר בקודקוד v. לפי הטענה ערך d[u] יהיה שווה בדיוק ל-|E'|- כיוון שכל קשת מ-E' תורמת 1- למסלול ~~[illegible]~~ וכל קשת אחרת לא תוסיף על משקל המסלול. המסלול הוא למעשה המסלול v→...→u→s שמצאנו. כיוון שהגרף מכוון, ואין מעגלים, אזי אין מעגלים שליליים (לפי הנחה G הוא גרף חסר מעגלים), ולכן לא יהיו ערכי d[] שהם ∞-.

נניח לא קיים מסלול כזה, אזי כל ערך d[] לכל קודקוד יהיה לכל הפחות |E'|+1-, כי לפחות קשת אחת מ-E' לא תהיה במסלול לכל קודקוד שהוא, ולכן נחזיר "לא".

סיבוכיות: B-F רגיל - $O(|E|\cdot|V|)$, מלבד הוספת s וקשתות ממנו לכל $u\in V$, מתן משקלים לקשתות ומעבר נוסף על כל $u\in V$ לבדיקת ערך d[] נקבל סיבוכיות הנ"ל של B-F, שהיא כלומר.

שאלה 2

יהא G=(V, E) גרף לא מכוון מיוצג ע"י רשימות שכנות, עם משקל שלם w(e) לכל קשת e∈E. לכל מספר שלם x, נסמן ב- $G^{(x)}=(V, E^{(x)})$ את תת הגרף המורכב מכל הקשתות שמשקלן לכל היותר x.
תארו אלגוריתם יעיל ככל האפשר המחשב את ה-x המקסימלי עבורו $G^{(x)}$ חסר מעגלים.

יעילות: $O(|E|\log|V|)$ (קרוסקל)

10

אלגוריתם:

נשנה מעט את אלג' קרוסקל לבניית MST. נשפר "שלב" להיות בדיקת קשתות ממשקל k שלם כלשהו במהלך הטבעי (השלב בו האלג' בודק לכל e∈E במשקל k האם היא נכנסת לעפ"מ או לא).

בכל שלב נמשיך לאלג' כרגיל: אם נתקלים בקשת e במשקל k בשלב הנוכחי שסוגרת מעגל, כלומר לא נכנסת ל-MST, עוצרים את הבניה, ומחזירים את x להיות k-1 [illegible]

אם בשלב כלשהו כל הקשתות בשלב נכנסו ל-MST, נעבור לשלב הבא.

~~[illegible]~~

(*) סיום האלגוריתם בסוף הדף בנספח התחתון

הסבר: כיוון שהגרף לא מכוון, תת הגרף חסר מעגלים חייב להיות יער. כיוון ש-x הוא המגבלה על משקל כל קשת שנרצה להכניס לאותו יער, אז נרצה למצוא יער בו כל קשת הוא ממשקל x≤k, וכל קשת מ-E במשקל הזה נכנסת להגבלה ונמצאת בו. אם למשל קיימת קשת ממשקל k≤x שסוגרת מעגל - הקשתות מאותו משקל לא שונות, והגרף אינו יער, כי ישנו מעגל.

אם כן, ע"י שימוש בקרוסקל בניית MST, נוודא שבכל שלב כל הקשתות ממשקל מסוים k נכנסות ל-MST. אם קשת כלשהי סוגרת מעגל, אזי המשקל המקסימלי שניתן להכניס ליער ללא סגירת מעגל (שכן, הבדיקה היא שכל הקשתות משקל k יהיו בתת הגרף) הוא כמובן x<k, כלומר ~~[illegible]~~ k-1 (מכיוון שזה המשקל הקודם בבדיקת האלג'). אם הצלחנו לבנות MST ויש אז קשתות מחוץ ל-MST במשקל השלב האחרון, אז השלב האחרון לא שלם, ולכן נחזיר את כיוון ה-k-1. אם השלב האחרון כן שלם, נחזיר את המשקל של הקשת הכבדה ביותר בגרף פחות 1 (ו-k כמתואר בשלב). המשך

(*) ~~[illegible]~~

מס. מחברת 57

שאלה 3

נתון גרף לא מכוון $G=(V, E)$ המיוצג ע"י רשימות שכנות, וצומת $s\in V$. לכל קשת $e\in E$ יש משקל שלם חיובי $w(e)$. לכל מסלול π בגרף היוצא מ- s נגדיר את משקלו להיות המשקל המקסימלי של קשתב- π. תארו אלגוריתם יעיל ככל האפשר המחשב, לכל צומת $v\in V$, את המשקל המינימלי של מסלול מ-s ל-v.

יעילות: דייקסטרה (דרוש ערימת פיבונאצ'י).
$O(|V|\cdot\log|V| + |E|)$

אלגוריתם:

נשתמש באלג' Dijkstra למציאת מק"ב מ-s, אך נשנה את פונ' ה-relax באופן הבא:

```
relax (u, v, w)
    if d(v) > max ( d(u), w(u,v))
        then d(v) ← max( d(u), w(u,v))
             π(V) ← u
```

בסוף האלג' d יכיל לכל $u\in V$ את המשקל המינימלי הנדרש.

הסבר:

9.5

סה"כ הנכונות נובעת מנכונות אלג' דייקסטרה, רק נראה שפונ' ה-relax מוגדרת היטב: נסתכל על הקשת (u,v). כשאנו מבצעים relax עליה אז: אם המסלול ל-v המוגדר דרך קשת זו, משקל הקשת הכבדה ביותר קטן ממשקל הקשת הקלה ביותר במסלול הנוכחי ל-v, לא יתקיים רק אם d(v) יהיה משקל גדול יותר גם מ-w(u,v) וגם מ-d(u), ולכן בודקים בינהם max. אם הוא אכן גדול, אז העדכון נכון: לאחריו d(v) יכיל את ~~משקל~~ הקשת הכבדה ביותר במסלול החדש - או w(u,v) או משקל הקשת הכבדה ביותר עד u, הוא d(u).
השימוש בדייקסטרה במקרה זה מאפשר, אם כל הנתונים בגרף בכל המשקלים חיוביים (משקל מקס' אינו מצטבר, ~~...~~ ולכן מעבר לשכנים שלהם לא משנים למשקל כל מסלול).

הערה: לא נמצא שינוי משמעותי, מהיחידות נראה כמו דייקסטרה המקורי.

קצת יותר להסביר מדוע $d = \delta$ בסיום

מס. מחברת 59

שאלה 4

נגדיר את y^i להיות השכפול של המחרוזת y עם עצמה i פעמים. למשל, $(ab)^3 = ababab$. נגדיר את כמות השכפולים במחרוזת x להיות r אם $x=y^r$ עבור איזושהי מחרוזת y ו- r הוא מקסימלי עם תכונה זו (כלומר y מינימלי).

תארו אלגוריתם יעיל ככל האפשר שמקבל כקלט מחרוזת $P[1…n]$ ומחשב לכל רישא $P_i=P[1…i]$ את כמות השכפולים ב- P_i. נתחו את זמן הריצה של האלגוריתם.

יעילות: $O(n^3)$

(הסבר למטה בפירוט)

אשר $O(n)$

אלגוריתם: האלגוריתם רץ איטרטיבית על i מ-1 עד n, ובמהלך מחשב כמות השכפולים של i מסוים, נקרא פונקציה זו $c(i)$, על חישובים קודמים באופן הבא:

$c(1)$ יהיה 1, תמיד. לכל i החל מ-2 נעשה:

יהי j אינדקס מ-i-1 יורד עד 1. נבדוק אם: אם $\frac{i}{j}$ שלם, נבדוק בעזרת kmp את הקלט $P[1,i]$ עם התבנית $P[1,j]$. ע"י שמירת ה-$\sigma(j)$, נבדוק אם $\sigma(j),\sigma(2j)...$ הם כולם שווים ל- j. אם כן נעצור ונחזיר לאותו i: $c(i):=\frac{i}{j}\cdot c(j)$.

אם לכל j לא עצרנו כלל, אזי נחזיר לאותו i: $c(i) = 1$.

נעשה כאמור את החישוב לכל i מ-2 עד n, והפונ' c תחזיר את הערך הנדרש.

הסבר: נניח עצרנו עבור i ומצאנו j כך ש- $\frac{i}{j}$ שלם. רק במקרה הזה יש פוטנציאל לשרשור $P[1,j]$ להיות משוכפל ב-$P[1,i]$. ע"י kmp נבדוק ב- $O(i)$, סה"כ $O(n)$, את ההתאמות. אם עצרנו כאמור ההתאמה הינה לדרוש: $\sigma(j)=\sigma(2j)=...=j$ אזי זה אומר ש- $P[1,i]=\underbrace{P[1,j]\circ...\circ P[1,j]}_{\frac{i}{j} \text{ פעמים}}$ - כנדרש, ולכן נחזיר $c(i):=\frac{i}{j}\cdot c(j)$. ואם היתה תת-מחרוזת קטנה יותר המינימלית לרישא, כבר חישבנו לאור כאשר חישבנו את $c(j)$.

אם לכל j לא עצרנו כלל, זה הרישא היחיד המתאים הוא $P[1,i]$, ולכן $c(i)=1$.

יעילות: לכל $i \le n$ מבצעים ריצה על j - עד כה $O(n^2)$. מספר הפעמים שמבצעים kmp כתלות [illegible] של i הוא (מספר המחלקים של i, 1,...,i) [illegible] i. [illegible], נסמן סדר גודל זה כ- $f(n)$

ורק זמן לומר שמבצעים $f(n)$ ריצות kmp, ולכן היעילות הוא $O(n^2\cdot f(n))$

n אחד: ריצות על i

n שני: ריצות kmp

מספר הסוגים [illegible] kmp

בהערכה גסה, אומר: $O(n^3)$

(כי $f(n) \le n$)

מס. מחברת 59

שאלה 5

נתון גרף מכוון $G=(V,E)$ על ידי רשימות שכנויות, ומשקל $w(e)$ חיובי לכל $e\in E$. הגרף מייצג רשת תקשורת, כאשר כל צומת מייצג משתמש, וכל קשת מייצגת קשר ישיר בין שני משתמשים. משתמש u יכול לשלוח הודעה למשתמש v אם קיים מסלול (מכוון) מ- u ל- v בגרף. משקל הקשת $(u,v)\in E$ הוא עלות הניתוק של הקשר הישיר בין המשתמשים u ו- v. כל צומת $v\in V$ מסווג כ"בוגר", "קטין", או "לא ידוע".

מטרתנו היא לחשב תת-קבוצה E' של קשתות הגרף כך שאם נמחק את כל קשתות E' מהגרף, לא יהיה מסלול משום צומת "קטין" לשום צומת "בוגר". תארו אלגוריתם יעיל ככל האפשר המקבל כקלט את G, w, ואת סיווג הצמתים, ומחזיר קבוצת קשתות E' המקיימת את התכונה לעיל, בעלת משקל כולל מינימלי מבין כל הקבוצות המקיימות את התכונה.

יעילות: דיניץ (או 0/1)

$O(|E|\cdot|V|^2)$

אלגוריתם:

נבנה רשת זרימה מהגרף G: נוסיף צומת s (מקור) ולכל $u\in V$ קטין נוסיף קשת מכוונת (s,u) במשקל ∞; נוסיף צומת t (בור) ולכל $v\in V$ בוגר נוסיף קשת מכוונת (v,t) במשקל ∞.

נשתמש במשקלות כקיבולים, ונמצא זרימה מקסימלית. נקח את הרשת השיורית האחרונה בה t לא נגיש מ-s, וניקח את החתך (S,T) כך ש-S הוא קבוצת הצמתים הנגישים מ-s, ו-T המשלימה שלה. נחזיר כ-E' את הקשתות בגרף המקורי המחברות צמתים מ-S לצמתים מ-T.

הסבר: בשלב הראשון ... הוא למצוא את ה-min cut; לפי משפט min cut-max flow, הזרימה המקסימלית שווה לחתך המינימלי. במקרה זה הזרמנו לתוך B הקטינים ∞, והוספנו מכל הבוגרים להזרים ∞ ל-t. לכן, הזרימה שתגיע ל-t תהיה לכל היותר הזרימה בחתך במשקל המינימלי בין הזרמת הזרימה מהקטינים לבוגרים, כלומר משמעה כאן שבמסלול מקטין לבוגר, ולכן אם נמחק את קשתות החתך, נקבל שהקשתות שמחקנו הן E' הרצויה - שמנתק כל קטין מכל בוגר. מה שנשאר נוכיח, שגם גודל E' הוא משקל מינימלי, נראה את החתך המינימלי, וכמובן שמאפשר הזרימה המקסימלית. לפיכך, אם נמצא את החתך האחרון שמושג ע"י ברשת השיורית לפי אלג' דיניץ, כיון שהוא התחום ומגיע לזרימה מקסימלית, הוא החתך המינימלי $\Rightarrow$ קבוצת הקשתות המחברות את החתך (מ-S ל-T, לא הפוך) הוא בעלת המשקל המינימלי, כנדרש.

יעילות: דיניץ - $O(|E|\cdot|V|^2)$.

שאלה 6

נתונה קבוצה E של n אינטרוולים על הישר $[a_1,b_1],\ldots,[a_n,b_n]$ כשלכל אינטרוול $[a_i,b_i]$ יש משקל חיובי w_i. תארו אלגוריתם יעיל ככל האפשר שמחשב תת-קבוצה E' של E עם משקל כולל מקסימלי כך שכל האינטרוולים ב-E' זרים בזוגות (החיתוך של כל שניים ריק). (**רמז**: סדרו את האינטרוולים בסדר עולה של נקודות הקצה הימניות שלהם, וקבלו נוסחת נסיגה עבור OPT(j), האופטימום עבור j האינטרוולים הראשונים.)

יעילות:

$O(n\cdot\log n)$
(בגלל המיון)

אלגוריתם: נמיין את n האינטרוולים לפי נק' הקצה הימניות שלהם בסדר עולה. נתייחס לאינדקס j כמו כאינדקס לפי אותו מיון. לכל $n\geq j\geq 1$ נחזיק שתי קבוצות:
$OPT^+(j)$ – הקבוצה E' עד האינטרוול $[a_j,b_j]$ הכוללת אותו; $OPT^-(j)$ – הקבוצה E' עד האינטרוול $[a_j,b_j]$ שאינה כוללת את $[a_j,b_j]$. נחשב איטרטיבית באופן הבא:

$OPT^+(1) := [a_1,b_1]\ ;\ OPT^-(1) := \phi$

חישוב ל-j+1 בהינתן של j:

~~[illegible]~~ if $a_{j+1} > b_j$ ---- אין חיתוך

$OPT^+(j+1) := OPT^+(j) \cup [a_{j+1},b_{j+1}]\ ;\ OPT^-(j+1) := OPT^+(j);$

if $a_{j+1} \leq b_j$ ---- יש חיתוך

(1) if $w[a_j,b_j] \geq w[a_{j+1},b_{j+1}]$:

$OPT^+(j+1) := OPT^+(j)\ ;\ OPT^-(j+1) := OPT^-(j)\ ;$

(2) if $w[a_j,b_j] < w[a_{j+1},b_{j+1}]$:

$OPT^+(j+1) := OPT^-(j) \cup [a_{j+1},b_{j+1}]\ ;\ OPT^-(j+1) := OPT^+(j)$

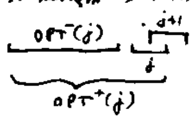
הסבר: נחשב זאת עד n. נחזיר: אם $w(OPT^-(n))$ גדול מ-$w(OPT^+(n))$ נחזיר את $OPT^-(n)$. אחרת נחזיר את $OPT^+(n)$.

נראה שהעדכון האיטרטיבי נכון:

• אם אין חיתוך, ברור: $OPT^+(j+1)$ ~~[illegible]~~ יכלול את $[a_{j+1},b_{j+1}]$ בוודאות. כ-$OPT^-(j+1)$, נרצה את הקבוצה האופטימלית שלא תכלול את $[a_{j+1},b_{j+1}]$, לכן זו תהיה $OPT^+(j)$, שהיה האופטימלי של j הקודם.

• אם יש חיתוך, לא ברור מספיק. כדי לשמור על אופטימליות עד סוף הריצה, אנו לא יודעים אם הוספת $[a_{j+1},b_{j+1}]$ (והורדת קודמים כדי למנוע חיתוך) תהיה תקינה לעדכון הבא, ולכן:

מקרה לדוגמה:



– אם אני במקרה (1), נרצה לשמור בכל מקרה על הדרכים כהן, כי $[a_{j+1},b_{j+1}]$ אינו משפר

אם אני במקרה (2), נרצה לשמור שיפור הכולל את $[a_{j+1},b_{j+1}]$ – האופטימלי יהיה להוריד את האינטרוולים ש"מפריעים" ולחבר את האופטימלי שנותר עם I, ולכן $OPT^+(j+1)=OPT^-(j)\cup I_{j+1}$

$I_{j+1} = [a_{j+1},b_{j+1}]$

ונרצה גם לשמור שיפור שלא כולל אותו, למקרה שנצטרך לו בהמשך, ואז יהיה זה האופטימלי עד I_{j+1} שלא כולל אותו ⇐ לכן $OPT^-(j+1)=OPT^+(j)$.

סיבוכיות: מיון – $O(n\log n)$

בסוף כל שלב מחזירים את השלב מבין השניים ע"י השוואת משקלם. זה לאחר – ליניארי $O(n)$. לכן ⇐

מס. מחברת 59

מסגרת חירום 1

המשך לשאלה ב

יעילות:
כבדף המקורי

אלגוריתם:

המשך האלג':

אם סיימנו לבנות MST, נבדוק האם יש עוד קשתות ממשקל השלב האחרון שלא נכנסו ל-MST. אם יש, אז הן יסגרו מעגל - אז נחזיר x להיות k-1 (k משקל השלב האחרון, ~~[illegible]~~).

אחרת, יהי j המשקל הבא אחרי משקל השלב האחרון, אז נחזיר x להיות j-1.

הסבר:

המשך ההסבר (שים לב: תחילת ההסבר בדף המקורי).

לאחר מבנית ה-MST שאני בונים למעשה את הגרף בו אין מעגלים, כי האלג' מתחשב לכל משקל בו נשקל בכל הקשתות מאותו משקל, ולכן להגדיר x רק שלא "מקלקל" בניית הגרף. אם הצלחנו לבנות MST, לא יעזור שנמשיך כי MST הוא עץ - וכל תוספת אליו תסגור מעגל.

יעילות - לכל היותר ~~[illegible]~~ $G^x = G$ ואין בעיות כלומר אלג' קרוסקל (השינויים לא משנים לסיבוכיות) - $O(|E|\log|V|)$