מטלה: 14 קורס: (20417) אלגוריתמים

שאלה 1

נגדיר גרף מכוון $G=(V,E)$ עם משקלות שבו קבוצת הקודקודים V מייצגת את n החברות שהחברה שלנו סוחרת במניותיהן. עבור כל שתי חברות $i \neq j$ שיש בינהן יחס מסחר, נוסיף קשת מכוונת (i,j) ונקבע את עלותה להיות $-\log r_{ij}$.

על מנת להבין את האינטואיציה מאחורי בחירת פונקצית משקל לוגריתמית בערכי יחסי המסחר, נבחן מהי בעצם מטרתינו. אנו מעוניינים למצוא מעגל C ב- G המהווה מעגל הזדמנות, כלומר, מעגל המקיים $\prod_{(i,j)\in C} r_{ij} > 1$. ראינו בסעיף 6.10 בע״מ 323-327 בספר אלגו׳ שזמן ריצתו $O(nm)=O(n^3)$ המוצא מעגלים שליליים בגרף מכוון. אם נבחר פונקציית משקל לוגריתמית בערכי יחסי המסחר, נוכל לבטא את התנאי הנ״ל ע״י **סכום הקטן מ-0** וזהו בדיוק מתאים למציאת מעגל שלילי בגרף מכוון.

אכן, מעגל C ב- G הוא מעגל הזדמנות אס״ם $\prod_{(i,j)\in C} r_{ij} > 1$. נוציא לוגריתם משני האגפים ונקבל ש- C הוא מעגל הזדמנות אס״ם $\sum_{(i,j)\in C} \log r_{ij} > 0$ או $\sum_{(i,j)\in C} -\log r_{ij} < 0$, כלומר, אס״ם C הוא מעגל שלילי. נוכל למצוא כזה (אם קיים) בזמן פולינמיאלי, כנדרש.

שאלה 2

נתון גרף לא מכוון $G=(V,E)$ עם משקלים ממשיים ושני צמתים $u,v \in V$. האלגוריתם שאתאר להלן (מבוסס על אלגוריתם בלמן-פורד המופיע בע״מ 315) מקבל כקלט גרף **מכוון** G' עם משקלים ממשיים ומחזיר את מספר המסלולים הקצרים ביותר בין u ל- v ב- G'. ניתן להפוך את G לגרף מכוון ע״י הפיכת כל קשת $e=(x,y)\in E$ לקשת מכוונת והוספת קשת בכיוון ההפוך – (y,x) – עם אותו משקל. ברור שכל מסלול המחבר בין u ל- v בגרף הלא-מכוון הוא גם מסלול המחבר בינהם בגרף המכוון (והם בעלי אותו משקל), ולהפך.

נסמן ב- $OPT(i,s)$ את העלות המינימלית של מסלול מ- u ל- s הכולל **בדיוק** i קשתות וב- $N(i,s)$ את מספר המסלולים האלה ונחשבם במקביל בעזרת הנוסחאות הרקורסיביות הבאות:

- אם $i=0$, אזי,

$$OPT(i,v)=0,\ N(i,v)=1 \quad and \quad OPT(i,s)=\infty,\ N(i,s)=0\ \forall s\neq v$$

- אם $0<i\le n-1$, אזי קודם מחשבים את

$$OPT(i,s)=\min_{w\in V}\left\{OPT(i-1,w)+c_{ws}\right\}$$

ואחרי שערך זה מתקבל מחשבים את

$$N(i,s)=\sum_{w:OPT(i,s)=\min_{w\in V}\left\{OPT(i-1,w)+c_{ws}\right\}} N(i-1,w)$$

הסבר: אם P מסלול אופטימלי המייצג את $OPT(i,s)$, אזי עלותו שווה לעלות המסלול המינימלי מ- u ל- w הכולל בדיוק $i-1$ קשתות ועוד עלותה של הקשת האחרונה (w,s). כמה מסלולים אופטימליים מ- u ל- s הכוללים i קשתות בדיוק קיימים? ניתן לענות על כך רק לאחר שנקבע עלותו של מסלול אופטימלי כזה. צומת $w\in V$ שהביא את $OPT(i-1,w)+c_{ws}$ למינימום מייצג מסלולים אופטימליים בעלי i קשתות המסתיימים בקשת (w,s). כמה מסלולים כאלה קיימים? $N(i-1,w)$.

נסמן את העלות המינימלית של מסלול מ- u ל- v ב- $OPT(v)$ (מסלול כזה כולל בהכרח לכל היותר $n-1$ קשתות) ואת מספר המסלולים האלה ב- $N(v)$, אזי $OPT(v)=\min_{0\le i\le n-1}\left\{OPT(i,v)\right\}$ ו- $N(v)=\sum_{OPT(v)=OPT(i,v)} N(i,v)$. $N(v)$ הוא הערך הרצוי.

זמן ריצה: זהה לזה של אלגוריתם בלמן-פורד $O(nm)$.

<u>שאלה 3</u>

בהינתן גרף מכוון $G=(V,E)$ עם פונקציית משקל $w:E\to\mathbb{R}$, צומת $t\in V$ ופונקי' $d:V\to\mathbb{R}$, האלגוריתם בעל העלות $O\left(|V|+|E|\right)$ שלהלן בודק אם d היא פונקציית מרחקים קצרים ל- t.

תחילה, נשים לב לעובדה הבאה: בהינתן קשת $e=(u,v)\in E$, אם מתקיים $d(u)>d(v)+w(e)$, אזי d לא יכולה להיות פונקציית מרחקים קצרים ל- t. מדוע הדבר נכון? נניח בשלילה את ההפך ויהי

P מסלול קצר ביותר מ- v ל- t. אורכו של P הוא $d(v)$. נתבונן במסלול מ- u ל- t המתחיל בקשת e וממשיך על P. אורכו של מסלול זה הוא $d(v)+w(e)$ והגענו לסתירה להגדרת d.

תאור האלגוריתם: נגדיר גרף מכוון $G'=(V,E')$ שיאותחל להיות G. נעבור על כל הקשתות $e=(u,v)\in E$. מהאמור לעיל, אם $d(u)>d(v)+w(e)$, אזי ניתן לעצור עם הפלט: "לא". אחרת, נשאיר ב- G' רק את הקשתות עבורן מתקיים $d(u)=d(v)+w(e)$ (כלומר, נסלק את אלה עבורן $d(u)<d(v)+w(e)$). כפי שאוכיח עוד רגע, d היא פונקציית מרחקים קצרים ל- t אס"ם לכל $u\in V$ קיים מסלול מ- u ל- t ב- G'. בהנחה שטענה זו נכונה, נריץ DFS מ- t על הגרף G' (הקשתות יפורשו כקשתות לא מכוונות). אם עבור $u\in V$ כלשהו, לא קיים מסלול מ- u ל- t ב- G' (DFS יגלה זאת), נעצור ונחזיר "לא". אחרת, כלומר לכל $u\in V$ קיים מסלול מ- u ל- t ב- G', נחזיר "כן". זמן הריצה האסימפטוטי של האלגוריתם הוא כזמן הריצה של DFS, היינו $O\left(|V|+|E|\right)$.

<u>טענה</u>: בסימונים לעיל, d פונק' מרחקים קצרים ל- t אס"ם לכל $u\in V$ יש מסלול מ- u ל- t ב- G'.

<u>הוכחה</u>: ($\Leftarrow$) יהא $u\in V$. נסמן ב- P את המסלול הקצר ביותר מ- u ל- t ב- G ונראה ש- P כלול גם ב- G'. תהי $e=(u,v)$ הקשת הראשונה על המסלול P. מההנחה שלנו, אורכו של P הוא, כמובן, $d(u)$ ואורכו של תת-המסלול $v-t$ של P הוא $d(v)$. לכן, $d(u)=d(v)+w(e)$ ומהגדרת G' נובע ש- $e\in E'$. באותו אופן, כל יתר הקשתות על המסלול P שייכות ל- E'.

($\Rightarrow$) נניח $d':V\to\mathbb{R}$ פונקציית המרחקים הקצרים ל- t. צריך להראות ש- $d'(u)=d(u)$ לכל $u\in V$. מההנחה שלנו, לכל $u\in V$ יש מסלול מ- u ל- t ב- G' (ולכן גם ב- G). מהגדרת G', אורכו של מסלול זה הוא $d(u)$ ולכן $d'(u)\le d(u)$. נניח בשלילה שקיים $u\in V$ עבורו $d'(u)<d(u)$. יהא P המסלול הקצר ביותר מ- u ל- t ב- G ו- $e=(x,y)\in E$ הקשת האחרונה בו עבורה $d'(y)=d(y)$ אך $d'(x)<d(x)$. כל תת-מסלול של P הוא מסלול מינימלי ולכן $d'(x)=d'(y)+w(e)$. בסך הכל קיבלנו ש- $d(x)>d(y)+w(e)$. ניזכר שתנאי זה עבור אחת מקשתות G היה אמור לגרום לאלגוריתם לסיים כבר בשלב יצירת G' וזה לא מה שקרה, לכן ההנחה $d'(u)<d(u)$ שגויה ו- $d'(u)=d(u)$, כנדרש.

שאלה 4

יהי $G=(V,E)$ גרף מכוון עם משקלים $\{c_{vu}\in\mathbb{R}\}_{(v,u)\in E}$ על הקשתות ויהיו $s,t\in V$.

א. האלגוריתם הבא מחשב את המסלול הקצר ביותר בין s ל-t המכיל מספר מינימלי של קשתות (אם יש כזה). למעשה, כל שעלינו לעשות הוא להריץ את אלגוריתם בלמן-פורד עם השינוי הקטן הבא. כאשר באים לשחזר מסלול קצר ביותר בין s ל-t מהמערך $M[0\ldots n-1,V]$, כפי שמודגם בע״מ 316 בספר, אין מתחילים מהתא $M[n-1,s]$, אלא מהתא $M[i,s]$ בעל האינדקס i, $0<i\le n-1$, הקטן ביותר עבורו מתקיים $M[i,s]=M[n-1,s]$. אם $M[n-1,s]=\infty$ הרי שאין אף מסלול קצר ביותר בין s ל-t. מהגדרת המערך M, התנאי שנוסח לעיל מבטיח שאין מסלול קצר ביותר בין s ל-t המכיל $i-1$ קשתות או פחות, אבל יש כזה המכיל i קשתות.

זמן ריצה: כמוסבר בע״מ 316 בספר, זמן הריצה של שיחזור המסלול הוא $O(in)$ וזמן הריצה של בניית המערך M הוא $O(mn)$, לכן סך הכל האלגוריתם רץ בזמן $O(mn)$.

ב. האלגוריתם הבא מחשב את המסלול הקצר ביותר בין s ל-t מבין כל המסלולים בין s ל-t המכילים מספר מינימלי של קשתות (אם יש כזה). שוב, כל שעלינו לעשות הוא להריץ את החלק הראשון של אלגוריתם בלמן-פורד – המידע הדרוש כדי לקבוע קיום וזהות של המסלול המבוקש קיים לרשותנו במערך $M[0\ldots n-1,V]$. אם $M[n-1,s]=\infty$, הרי שאין אף מסלול קצר ביותר בין s ל-t. אחרת, נמצא את האינדקס i, $0<i\le n-1$, הגדול ביותר המקיים $M[i,s]\neq\infty$ וממנו נתחיל לשחזר מסלול קצר ביותר בין s ל-t בעל i קשתות לכל היותר (כפי שמוסבר בע״מ 316 בספר). למעשה, מהגדרת המערך M, מסלול זה כולל בדיוק i קשתות ואין אף מסלול בעל פחות מ-i קשתות בין s ל-t.

זמן ריצה: כמו ב-א', היינו, $O(mn)$.

שאלה 5

א. בהנתן מטבעות בערכים שלמים חיוביים $x_1, x_2, \ldots, x_n$ ושלם חיובי v, נבנה אלגוריתם תכנון דינמי הבודק האם ניתן להציג את v כסכום של מטבעות מתוך הקבוצה (כלומר, לפרוט את v).

נגדיר מערך דו-מימדי $M[0\ldots n, 0\ldots v]$ של ערכים בוליאניים כך שלכל i ו- j שלמים חיוביים, $0 \le i \le n$ ו- $0 \le j \le v$, $M[i, j] = true$ אס"ם ניתן להציג את j כסכום של i המטבעות הראשונים. אנו מעוניינים, כמובן, בערכו של $M[n, v]$.

כרגיל, נבנה את M ברקורסיה:

- לכל $0 \le i \le n$, נקבע $M[i, 0] = false$ ולכל $0 \le j \le v$, נקבע $M[0, j] = false$.
- אם $0 < i \le n$ וגם $0 < j \le v$, $M[i, j] = \begin{cases} M[i-1, j] & j < x_i \\ M[i-1, j] \vee M[i, j - x_i] & otherwise \end{cases}$.

הסבר: ברור שלא ניתן להשתמש במטבע ה- i-י אם $j < x_i$ ובמקרה כזה $M[i, j] = M[i-1, j]$. במידה ו- $j \ge x_i$ קיימת האפשרות שהמטבע ה- i-י **כן** משתתף באיזושהי פריטה של j ואז ערכו של $M[i, j]$ שווה ל- $M[i, j - x_i]$ או שהוא **לא** משתתף בפריטה כזאת ואז $M[i, j]$ שווה ל- $M[i-1, j]$ ואין אפשרויות נוספות. מכאן מתקבלת נוסחת הרקורסיה שלמעלה.

אלגוריתם הפותר את הבעיה הוא אם כן:

```
CHANGE(coins,v)
    n ← length[coins]
    Array M[0…n,0…v]
    for i ← 0 to n
        do M[i,0] ← false
    for j ← 0 to v
        do M[0,j] ← false
    for i ← 1 to n
        do for j ← 1 to v
                do Use the above recursion to compute M[i,j]
    return M[n,v]
```

זמן הריצה: לולאת ה-for בשורות 3-4 אורכת $\Theta(n)$, לולאת ה-for בשורות 5-6 אורכת $\Theta(v)$ ולולאות ה-for המקוננות בשורות 7-9 אורכות $\Theta(nv)$ משום שחישוב M[i,j] אורך זמן קבוע. בסך הכל זמן הריצה של האולגוריתם $CHANGE$ הוא $\Theta(nv)$.

ב. נניח כעת שמספר המטבעות שניתן להשתמש בהן הוא לכל היותר k. שינוי זה מצריך הוספת מימד נוסף למערך M שיבטא את מספר המטבעות המירבי בכל תת-פתרון.

ביתר פירוט, נגדיר מערך תלת-מימדי $M[0\ldots n, 0\ldots v, 0\ldots k]$ של ערכים בוליאניים כך שלכל i, j ו- m שלמים חיוביים, $0 \le i \le n$, $0 \le j \le v$ ו- $0 \le m \le k$, $M[i, j, m] = true$ אם״ם ניתן להציג את j כסכום של לכל היותר m מ- i המטבעות הראשונים. אנו מעוניינים ב- $M[n, v, k]$.

כמו קודם, נבנה את M ברקורסיה:

- לכל $0 \le i \le n$ ו- $0 \le m \le k$, נקבע $M[i, 0, m] = false$, לכל $0 \le j \le v$ ו- $0 \le m \le k$ נקבע $M[0, j, m] = false$ ולכל $0 \le i \le n$ ו- $0 \le j \le v$ נקבע $M[i, j, 0] = false$.
- אם $0 < i \le n$ וגם $0 < j \le v$ וגם $0 < m \le k$,

$$M[i, j, m] = \begin{cases} M[i-1, j, m] & j < x_i \\ M[i-1, j, m] \vee M[i, j-x_i, m-1] & otherwise \end{cases}.$$

הסבר: דומה לסעיף א׳. יש לשים לב לעובדה שנדרשת הפחתה של 1 מ- m במקרה שמטבע i נכלל בפריטה מסויימת של j עבור $0 < m \le k$, $0 < j \le v$ ו- $0 < i \le n$.

להלן האלגוריתם $CHANGE'$ הפותר את הבעיה המעודכנת:

```
CHANGE'(coins,v,k)
n ← length[coins]
Array M[0…n,0…v,0…k]
for i ← 0 to n
    do for j ← 0 to v
        do M[i,j,0] ← false
for i ← 0 to n
    do for m ← 0 to k
        do M[i,0,m] ← false
for m ← 0 to k
    do for j ← 0 to v
        do M[0,j,m] ← false


for i ← 1 to n
    do for j ← 1 to v
        do for m ← 1 to k
            do Use the above recursion to compute M[i,j,m]
return M[n,v,k]
```

זמן ריצה: בדומה לניתוח המפורט המופיע בסעיף א', קל לראות שזמן הריצה הוא $\Theta(nvk)$.