הפיתרון למועד א' של תשס"ט

1. א. כל קודקוד v יקבל שם (i,j), כאשר i זה מס' השורה בסריג שבה v ממוקם, ו-j מספר העמודה שבה v ממוקם.

בהינתן כתובת היעד (i',j'), קודקוד (i,j) ישלח את ההודעה באופן הבא:
אם $i < i'$, אז הוא ישלח את ההודעה ל-$(i+1,j)$.
אם $i > i'$, אז הוא ישלח ל-$(i-1,j)$.
אם $i = i'$ ו $j < j'$ אז הוא ישלח ל-$(i,j+1)$, ואחרת ל-$(i,j-1)$.

ברור שכל קודקוד משתמש ב-$O(\log n)$ זיכרון.
כמו-כן קל לראות באינדוקציה, שאורך מסילת הניתוב מ-(i,j) ל-(i',j') הוא בדיוק $|j'-j|+|i'-i|$, והביטוי הזה הוא בדיוק המרחק בין שני הקודקודים המתאימים בגרף.
(הביטוי הזה נקרא "מרחק מנהטן", או Manhattan distance.)

1.ב. קודקוד u שולח את ההודעה שלו לכל
שכניו. אם אחד מהשכנים הוא v, אז
v מודיע ל-u שהוא קיבל את ההודעה.
בכל מקרה, כ״א מהשכנים שולח ל-u
Ack עם ביט שאומר האם ~~הוא היעד~~ היעד
קיבל את ההודעה או לא.
אם u לומד ש-v קיבל את ההודעה,
הוא שולח ~~שהודעה הודעה~~ לכל שכניו הודעה
שאומרת שאפשר לסיים את ה-broadcast.
אחרת, הוא מודיע לכל שכניו שצריך לחפש
רמה נוספת.
באופן כללי, באיטרציה i ית׳, $i < dist(u,v)$,
נבנה כבר עץ BFS שמושרש ב-u
שעומקו הוא i. בתום האיטרציה u יודע
האם v קיבל כבר את ההודעה או לא.
אם כן אז u יוצא broadcast עם הודעת
הסיום על-פני העץ. אחרת u יוצא
broadcast עם הודעה שאומרת לגלות עוד רמה.
בצורה כזאת כעבור $dist(u,v)$
איטרציות, v מקבל את ההודעה,
והתהליך מסתיים.
ניתוח מס׳ ההודעות:
באיטרציה i, עץ ה-BFS שנבנה עד השלב
הזה מכיל $\Theta(i^2)$ קודקודים.
לכן באיטרציה i נשלחות $\Theta(i^2)$ הודעות.

לכן סה"כ ב $dist(u,v)$ איטרציות נשלחות

$$\sum_{i=1}^{dist(u,v)} \Theta(i^2) = \cancel{\Theta(n^3)}\ \Theta(dist^3(u,v))$$

הודעות.

2) א. 29 צבעים ניתנים לייצוג ע"י 5 ביטים. מפעילים את האלג' Cole-Vishkin

3 סיבובים
פעם אחת, ומקבלים $\lceil \log 5 \rceil + 1 = 4$
ביטים של "צבע".
מפעילים פעם נוספת ומקבלים 3 ביטים
של "צבע".
מפעילים עוד פעם ומקבלים 6 צבעים
אפשריים.

6 סיבובים
מבצעים את פעולת ה-shift-down
כדי להבטיח שלכל צומת u ושני בניו x
ו-y הצבעים של x ו-y שווים צ"ל.
אחר נפטרים מצבע 6.
שוב, עושים shift-Down ונפטרים מצבע 5.
" " " " 4.

סה"כ 9 סיבובים

ב. עץ בלי אוריינטציה הוא גרף בעל ~~loglog~~
יערות 1. עבור פרמטר q ~~$q = \log$~~
~~סדר $\varepsilon \geq 1$~~, מפרקים אותו ל-$q \cdot q = q$ יערות
בזמן $O(\log_q n)$ ~~$O(\frac{\log n}{\log\log n})$~~.

בהינתן הפירוק ל-q יערות, נחשב
3^q-צביעה של הגרף בזמן נוסף של $\log^* n$,
וב-3^q זמן נוסף נהפוך את הצביעה ל-MIS.
הזמן הכולל הוא $O(\log_q n + 3^q)$.
נציב $q = \frac{1}{2} \log\log n$ ונקבל $O(\frac{\log n}{\log^{(3)} n})$ זמן.

שאלה 3:

א. נפתרה בכיתה בשיעור האחרון.

ב. בעיית עדכון השכנים נלמדה בשיעורים.

שאלה 4:

נבנה עץ BFS עבור הגרף,
ונפעיל את האלג' Pipeline_MST, תוך
התייחסות רק לצלעות שמחברות בין
הפרגמנטים השונים בגרף.
בניית BFS תדרוש $O(Diam(G))$ זמן
Pipeline_MST תדרוש זמן של $O(Diam(G)+k)$,
כאשר k זה מספר הצלעות "החסרות"
כדי להשלים את הפירוק לפרגמנטים שיש לנו
לכדי MST של הגרף כולו.
נשאר להראות ש- $k=O(n^{1/3})$.
לצורך זה נראה שמס' הצלעות החסרות ~~[illegible]~~ שווה
למספר הפרגמנטים פחות 1.
(זה גורר את הטענה, כי מס' הפרגמנטים
הוא לכל היותר n חלקי הגודל המינימלי
של פרגמנט בודד, דהיינו $O(n^{1/3})=\frac{n}{\Omega(n^{2/3})}$)

יהיו $F_1, F_2, \ldots, F_\ell$ הפרגמנטים של MST
T שמכסים את כל קודקודי הגרף (וזרים
זה לזה). אם נכווץ כ"א מהפרגמנטים F_i
לקודקודים בודדים v_i, נקבל עץ חדש T'

שבו יש צלע (v_i, v_j) אמ"ם ב-T

יש צלע ~~שמחברת~~ שמחברת פרגמנט

F_i עם פרגמנט F_j. (T' הוא עץ.

כי אחרת היינו מקבלים סתירה לכך

ש-T הוא עץ)

מס' הצלעות ב-T' הוא בדיוק $l-1$.

כמו-כן מס' הצלעות ב-T' שווה בדיוק

למס' הצלעות ב T שמחברות פרגמנטים

שונים, וזה משלים את ההוכחה.

מ-ש.ל.

5.א. נצבע צלע אחרי צלע ע״י פלטת
הצבעים $\{1,2,\ldots,2\Delta-1\}$.
בבואנו לצבוע את הקשת $e=(u,v)$
יש לכל היותר $2\cdot(\Delta-1)$ צבעים אסורים
שהם צבעים של הצלעות הסמוכות ל-e.
לכן תמיד יהיה צבע פנוי עבור e.

ב. נחלק את כל צלעות הגרף ל-3
קבוצות. בקב' הראשונה E_1 יהיו הצלעות
המחברות קודקודים עם ID-ים זוגיים
ב-E_2 אי-זוגיים וב-E_3 הצלעות שמחברות
קודקוד עם ID זוגי לאי-זוגי.
נפעיל רקורסיה בכ״א מן הקבוצות הנ״ל,
ונקבל $3\cdot(2\Delta-1)=6\Delta-3$ צביעת-צלעות
לגרף. תוך $(6\Delta-3)-(2\Delta-1)=4\Delta-2$
סיבובים נוספים נצמצם את מס'
הצבעים ל-$(2\Delta-1)$ ע״י כך שניפטר
מצבע אחד בכל סיבוב.
הזמן הכולל הוא $(4\Delta-2)$ כפול מס' הרמות
של הרקורסיה, והוא כמובן $O(\log n)$.
(כי בכל הפעלה רקורסיבית אפשר "לזרוק" את
הביט הכי פחות משמעותי מה-ID-ים של
הקודקודים המשתתפים)
לכן סה״כ הזמן הוא $O(\Delta\cdot\log n)$.