שאלה 1

נסמן $\left\lceil \frac{n}{2} \right\rceil$=mid, ונעבור על כל איברי המערך תוך מציאת האיבר המקסימלי max והאיבר המינימלי min – O(n) ("יבלע בזמן הריצה הצפוי של 'Select).

נקרא לפונקציה Select'(min, max, mid) אותה נתאר כעת:

הפונקציה 'Select תעבור על איברי המערך ותמצא pivot המקיים min<pivot<max – O(n).

לאחר מכן 'Select תעבור שוב על איברי המערך ותסכום את מספר האיברים שקטנים מה pivot שלנו בעזרת counter – O(n). נשים לב כי pivot הינו האיבר ה counter+1 ביחס למערך.

*אם counter+1=mid נחזיר את pivot וסיימנו.

*אם counter+1>mid, אז נקרא באופן רקורסיבי ל **Select'(min, pivot, mid)**.

*אם counter+1<mid, אז נקרא באופן רקורסיבי ל **Select'(pivot, max, mid-(counter+1))** (שינוי הגבולות במקרה זה משנה את מיקומו של האיבר אותו אנו מחפשים כי מיקום איבר ה 0 גדל).

<u>זמן ריצה:</u> נראה כי נצפה שבכל קריאה רקורסיבית אנו מקטינים את הטווח פי $\frac{7}{8}$:

Good pivot =נמצא ברבע השני\השלישי של המערך ומקטין את הטווח פי לפחות $\frac{3}{4}$.

Bad pivot = נמצא ברבע הראשון\הרביעי של המערך ומקטין את הטווח בלפחות 1.

ולכן: P[Good pivot]=P[Bad pivot]=$\frac{1}{2}$

$$E[elementsNotForTheNextRange] \geq P[good\ pivot] * \frac{1}{4}n + P[bad\ pivot] * 1 = \frac{1}{2} * \frac{1}{4}n + \frac{1}{2} * 1 \geq \frac{1}{8}n$$

לכן נצפה כי הפונקציה הרקורסיבית 'Select מקיימת $T(n) = T(\frac{7}{8}n)$. ננתח בעזרת שיטת האיטרציה כמה פעמים נצפה כי 'Select תקרא לעצמה:

$$T(n) = T\left(\frac{7}{8}n\right) = T\left(\frac{7^2}{8^2}n\right) = \cdots = T\left(\frac{7^i}{8^i}n\right)$$

i הוא מספר הפעמים שהפונקציה קוראת לעצמה. נעצור ב T(1) ונקבל כי $\left(\frac{7}{8}\right)^i n = 1 \gg n = \left(\frac{8}{7}\right)^i \gg i = \log_{\frac{8}{7}} n$ ולכן מצאנו כי נצפה ש 'Select תקרא לעצמה O(logn) פעמים.

כעת נשים לב כי בכל קריאה לעצמה 'Select מבצעת פעולות ב O(n), ומכאן שזמן הריצה הצפוי של הפונקציה הינו O(n)*O(logn)=O(nlogn).

<u>תוספת זיכרון</u> – 'Select משתמשת ב O(1) זיכרון בכל פעם, נצפה כי תקרא לעצמה O(logn) פעמים ולכן נצפה כי הזיכרון הנוסף יהיה O(logn).

<u>שאלה 2</u>

נאתחל ערימת מינימום עם האיבר המינימלי של כל מערך, כאשר כל איבר מחזיק במפתח ובמספר המערך ממנו הגיע – **O(logk).**

נשלוף את איבר המינימום מהערימה – O(logk) ונשמור אותו במשתנה זמני המתאים למערך אליו שייך האיבר, נתקדם אל האיבר הבא במערך ממנו הגיע ונכניס אותו לערימת המינימום O(logk). כאשר כל המשתנים הזמניים המתאימים למערכים עודכנו ("נוצר הטווח הראשון"), נשמור את האיבר המקסימלי והאיבר המינימלי מבין המשתנים הזמניים, ואת גודל ההפרש ביניהם. נמשיך באותו האופן על כל n האיברים, ונעדכן את האיבר המקסימלי, האיבר המינימלי וההפרש ביניהם במידת הצורך. לאחר שסיימנו לעבור על כל האיברים, נחזיר את הטווח עבורו ההפרש הקטן ביותר התקבל לאורך הדרך.

<u>נכונות:</u> יהי (a,b) הטווח המינימלי הרצוי. במהלך ריצת האלגוריתם b יסומן מתישהו כאיבר המקסימלי מכיוון שלאחר שנשלף מערימת המינימום, כל האיברים שנשלפו לפניו קטנים ממנו. נבחין כי ברגע ש b הינו איבר המקסימום, קיים x שהינו איבר המינימום. נניח בשלילה כי a<x, אז (x,b) הדוק יותר ועומד בתנאי של טווח - סתירה. נניח בשלילה כי x<a, אז קיים איבר נוסף מהמערך של x בטווח (x,b). לפי האלגוריתם x היה מתקדם אל האיבר הנ"ל לפני ש b היה נשמר – סתירה. לכן x=a, אז הוכחנו כי במהלך האלגוריתם הטווח הרצוי (a,b) נלכד.

<u>זמן ריצה:</u> עבור כל אחד מתוך n האיברים נבצע מספר קבוע של פעולות המתרחשות בזמן של O(logk), ומכאן שזמן הריצה של האלגוריתם הינו O(nlogk).

<u>שאלה 3</u>

א. נבנה עץ AVL ריק שאינו מאפשר הכנסה של 2 מפתחות זהים, ומערך A בגודל n. נעבור על כל איברי הקבוצה S ונכניס אותם לעץ ולמערך (איבר יכנס למערך אך ורק אם הוכנס לעץ). לאחר שהוכנסו כל המפתחות, בעץ O(logn) איברים, ולפי מה שנלמד בכיתה כל פעולת הכנסה התרחשה ב O(loglogn) וכל פעולת חיפוש תתרחש ב O(loglogn). כעת נעבור על כל איברי המערך שבנינו. אם ki הינו המפתח של האיבר בתא ה i, נבדוק האם המפתח z-ki נמצא בעץ.

<u>זמן ריצה:</u> אתחול המערך מתרחש ב O(n). אנו עוברים על כל n האיברים ב S ומכניסים אותם לעץ, כאמור כל פעולת הכנסה מתרחשת ב O(loglogn), ולכן אתחול העץ קורה ב O(nloglogn) (הכנסת כל איבר למערך מתרחשת בזמן קבוע). לאחר מכן עבור כל אחד מ logn האיברים שבמערך, זמן החיפוש בעץ מתרחש ב O(loglogn) ולכן החיפוש קורה ב O(logn*loglogn). בסה״כ נשים לב כי כל האלגוריתם מתרחש ב O(n*loglogn).

ב. נבנה מערך ריק בגודל n. נעבור איבר איבר על איברי קבוצה S, נבדוק האם קטן מ z. במידה וכן, נכניס אותו למערך – O(n). לאחר מכן נמיין את המערך בעזרת Counting Sort כשאנו יודעים שאיברי המערך בטווח {0,...,z}, ועי״פ מה שנלמד בכיתה המיון מתרחש ב O(n+z)=O(n) (משום ש z<n). כעת נאתחל שני מצביעים min ו max לאיבר הראשון ואחרון במערך בהתאמה, ונשתמש באלגוריתם שנלמד בכיתה: אם min+max=z סיימנו. אם min+max<z, נקדם את min איבר אחד קדימה במערך. אם min+max>z, נעביר את max איבר אחד אחורה במערך. ניתן לראות כי החיפוש הנ״ל מתבצע ב O(n), ובסה״כ כל האלגוריתם מתואר על ידי מספר קבוע של פעולות שכל אחת מתרחשת ב O(n), אז האלגוריתם כולו רץ ב O(n).

שאלה 4

נבנה גרף $G'=(V', E')$ באופן הבא: לכל v ששייך ל V ניצור שני קודקודים v.e ו v.o ב V', ולכל קשת (v, u) ב E ניצור שתי קשתות $(v.e, u.o)$ ו $(v.o, u.e)$ ב E'. נשים לב כי $|V'|=2|V|$ ו $|E'|=2|E|$. על מנת למצוא את המרחק הקצר ביותר של קודקוד p כלשהו מ s המכיל מספר זוגי של קשתות, נפעיל את אלגוריתם BFS על s.e, והמרחק הנ"ל הינו המרחק שמתקבל מאלגוריתם ה BFS עבור p.e.

<u>נכונות:</u> מכיוון שבכל צעד מקודקוד v לקודקוד u אנו עוברים מ v.e ל u.o (או להפך) על מנת להגיע מ s.e ל p.e היה עלינו לעשות מספר זוגי של צעדים, ולכן המסלול שמצאנו מ s ל p הינו זוגי. מינימליות המסלול נובעת ישירות מאלגוריתם ה BFS.

<u>זמן ריצה:</u> אתחול הגרף G' קורה ב $O(|V'|+|E'|)=O(|V|+|E|)$. בנוסף, לפי מה שלמדנו בכיתה זמן הריצה של BFS הינו $O(|V'|+|E'|)=O(|V|+|E|)$. ולבסוף שליפת הערך מ p.e הינו $O(1)$, ובסה"כ זמן הריצה של האלגוריתם הינו $O(|V|+|E|)$.

שאלה 5

א. נשים לב לאבחנה הבאה: במידה וקיים שורש בגרף G, בכל ריצת DFS שתתחיל מקודקוד כלשהו בגרף, הקודקוד בעל זמן הסיום הגדול ביותר הינו שורש בגרף. זאת משום שבאלגוריתם DFS קודקוד משנה את צבעו לשחור ומעדכן את זמן הסיום שלו, רק לאחר שלא ניתן עוד להגיע ממנו לקודקודים שהסריקה הנוכחית לא ביקרה בהם עדיין. לכן, במהלך הסריקה כאשר הגענו לקודקוד שהינו שורש, הוא יהפוך שחור רק לאחר שסיימנו לבקר בכל שאר הקודקודים בגרף מעצם הגדרתו כשורש.

לכן, על מנת למצוא קודקוד שהינו שורש, נבצע סריקת DFS מקודקוד כלשהו בגרף, ואז לבצע סריקת DFS נוספת מהקודקוד האחרון שהפך לשחור בסריקה הקודמת (בעל זמן הסיום הגדול ביותר). במידה וגם בסריקה הנוכחית הוא ישנה אחרון את צבעו לשחור, הוא שורש ונחזירו. אחרת נודיע כי אין שורש בגרף.

<u>זמן ריצה:</u> נשים לב כי במהלך האלגוריתם ביצענו שתי סריקות DFS, לכן על פי מה שנלמד בכיתה זמן הריצה של האלגוריתם הינו $O(|V|+|E|)$.

ב. נשים לב לאבחנה הבאה במידה וקיים שורש בגרף: קודקוד נוסף הוא שורש אם ורק אם יש מסלול ממנו אל השורש עליו ידוע.

$\Leftarrow$ באופן טריויאלי מהגדרת שורש.

$\Rightarrow$ אם יש מסלול מקודקוד כלשהו לקודקוד שהוא שורש, אז ניתן להגיע ממנו אל כל הקודקודים בגרף.

לכן, במידה וקיים שורש כלשהו, נמצא אותו על פי האלגוריתם שתואר בסעיף א' - $O(|V|+|E|)$. לאחר מכן נעבור על כל קשתות הגרף ונשנה כל קשת $\{u,v\}$ לקשת $\{v,u\}$ – $O(|E|)$. לבסוף, נבצע סריקת BFS מהשורש שמצאנו - $O(|V|+|E|)$, וכל קודקוד שנצליח להגיע אליו הוא קודקוד שקיים מסלול ממנו אל השורש הידוע, ועל פי האבחנה הוא שורש בעצמו אז נחזיר אותו.

<u>זמן ריצה:</u> ביצענו שתי פעולות ב $O(|V|+|E|)$ ופעולה אחת ב $O(|E|)$, ומכאן שזמן הריצה של האלגוריתם הינו $O(|V|+|E|)$.

שאלה 6

נשתמש בשיטת הצבירה שנלמדה בכיתה. נביט בטבלא הבאה, כאשר f(m) הינו זמן הריצה של פעולת האלגוריתם עבור הקלט ה m לפי הנתון:

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...
f(m)	1	2	1	4	1	1	1	8	1	1	1	1	1	1	1	16	...

נשים לב כי $\sum_{m=1}^{n} f(m) \leq n + \sum_{i=1}^{logn} 2^i$ מכיוון שחזקה של 2 יכולה להופיע לכל היותר logn פעמים וסכמנו 1״ים גם עבור הפעולות שנסכמו כבר כחזקה של 2. ע״פ סכום של סדרה הנדסית $\sum_{i=1}^{logn} 2^i = 2n - 2$ ולכן:

$$\sum_{m=1}^{n} f(m) \leq n + \sum_{i=1}^{logn} 2^i = n + 2n - 2 \approx 3n$$

מכאן שזמן הריצה הכולל של האלגוריתם עבור n קלטים הוא O(3n), על כן העלות לשיעורין של כל פעולה היא $\frac{O(3n)}{n} = O(1)$.