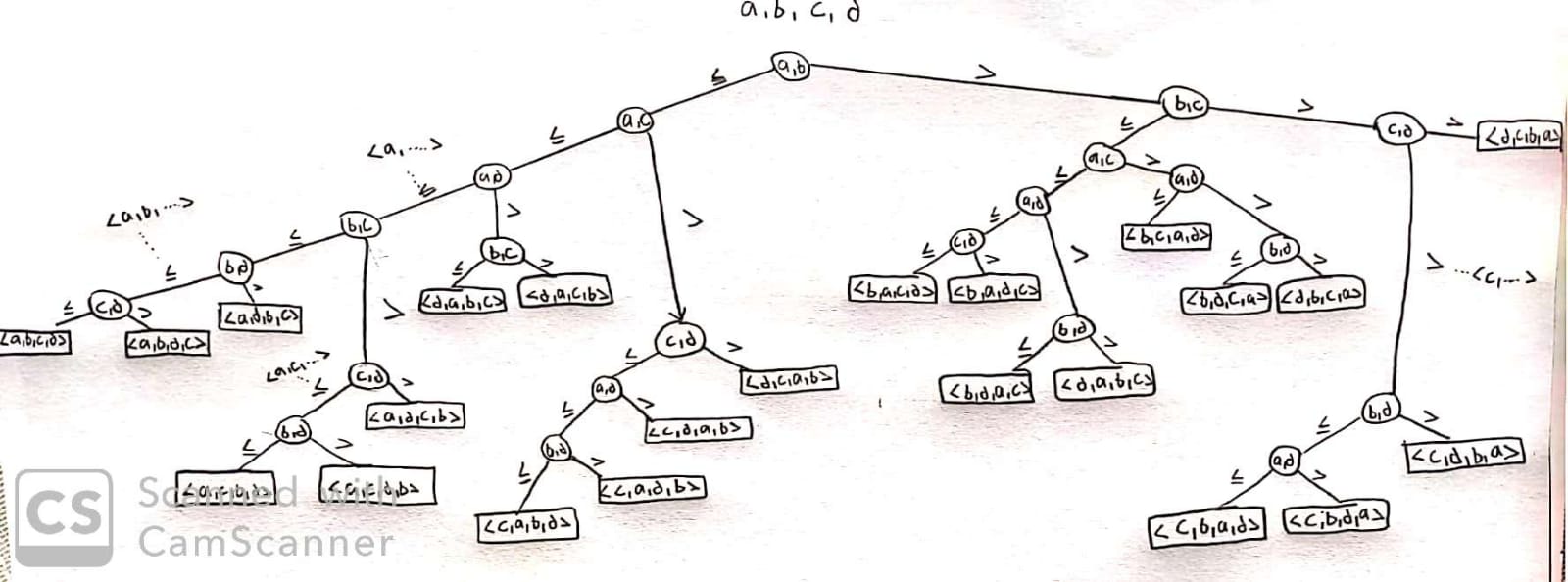
**שאלה 1:**

1. 

מצורף קובץ מקורי

1. על פי ההסבר בשאלה ז-2 במדריך הלמידה, בעץ החלטה של אלגוריתם מיון גובה העץ שווה למסלול הארוך ביותר כלומר גובה העץ שווה למספר ההשוואות שהאלגוריתם מבצע במקרה הגרוע. במיון ערימה האלגוריתם מבצע במקרה הגרוע השוואות, ולכן זהו גם גובה עץ ההחלטה המתאים.
2. על פי התשובה לשאלה ז-4 במדריך למידה, בכל אלגוריתם מיון השוואת כל אחד מn האיברים חייב להשתתף לכל הפחות בהשוואה אחת, ולכן חייבות להתקיים לפחות n-1 השוואות, ולכן אורך המסלול הקצר ביותר בין השורש לעלה האפשרי בעץ החלטה של אלגוריתם מיון מבוסס השוואות בעץ ההחלטה מהשורש לעלה יהיה לפחות .n-1
3. שאלה דומה לשאלה מממן 12, בעזרת בדיקה מקדימה האם המערך כבר ממוין, בדיקה זו כוללת בדיוק n-1 השוואות.

**שאלה 2:**

1. כן, ממשפט 8.2 כל אלגוריתם מיון השוואה עורך השוואות במקרה הגרוע, ולאחר מכן במסקנה 8.2 הוכח שמיוני ערמה ומיזוג אופטימליים, כלומר עורכים השוואות, ולכן ניתן לקחת כל מיון מבוסס השוואת ולעשות אותו יותר "גרוע" – רק כדי להגיע לאלגוריתם שזמן הריצה שלו במקרה הגרוע יהיה בין היתר על ידי הוספת פעולות/לולאות שאינן משנות דבר – לדוגמא הוספת לולאה קבועה פנימית שרצה פעמים.

ובנוסף לשלב את מה שהראנו בשאלה 1 ד' לעיל, שאפשר לשפר כל אלגוריתם מיון מבוסס השוואות שזמן ריצתו במקרה הטוב יהיה לינארי.

ולכן הטענה תיתכן.

1. לא, טענה זו היא בדיוק הטענה שמוכחת בתרגיל 8.1-3 בספר (שניתנת לו תשובה בשאלה ז-6 במדריך הלמידה)

**שאלה 3:**

1. האלגוריתם Counting Sort מעמוד 140 בספר עונה לקריטריונים, זמן הריצה שלו הוא , והוא נחשב יציב לפי שאלה ז-7 עמוד 116 במדריך הלמידה
2. ניתן להשתמש באלגוריתם סביב הpivot , ולמעשה המערך יסודר כל שכל האיברים שקטנים או שווים ל0 ימוקמו בחלק השמאלי וכל האיברים שגדולים מ0 ימוקמו בחלק הימני, ומבחינתנו עם ערכים של 0 ו1 בלבד, המערך ימוין במקום, זמן הריצה יהיה .
3. ניתן להשתמש באלגוריתם שממיין במקום, ונחשב יציב – לפי תרגיל 8.3-2 בספר – שאלה ז-14 עמוד 116 במדריך הלמידה.

**שאלה 4:**

הערה: בשאלה הזו רשמתי פסאודו קוד מפורט רק לשגרות שנראו לי בעלות לוגיקה כל שהיא, לפי ההודעה של אורי בפורום הבנתי שכלל לא חייב להוסיף פסאודו קוד.

בכל שלושת הסעיפים  *פשוט יוצר רשימה חדשה שתחילתה מאותחלת בNIL ומיד מוחזרת*

*כמובן שזמן הריצה כאן קבוע והוא*

1. *שגרת :*

*מעבר לינארי על כל איברי הרשימה עד למציאת האיבר הכי קטן שגדול מהאיבר שיש להכניס, והכנסת האיבר במקום מיד אחרי האיבר שמצאנו.*

*זמן ריצה:*

*שגרת :*

*הוצאת האיבר הראשון (בראש) הרשימה המקושרת הממוינת, שהוא המינימלי, ועדכון ראש הרשימה לאיבר הבא.*

*זמן ריצה:*

*שגרת :*

*החזרת הערך שנמצא באיבר הראשון ברשימה המקושרת הממוינת שהוא המינימלי.*

*זמן ריצה:*

*שגרת :*

*אתחול רשימה ריקה חדשה ומעבר במקביל על 2 הרשימות עד שמגיעים לסוף של שניהם, עם 2 מצביעים שבהתחלה מאותחלים לראש כל רשימה, בכל איטרציה בדיקה איזה איבר נוכחי מאיזה רשימה קטן יותר, והכנסה שלו לרשימה החדשה עם עדכון המצביע של הרשימה שממנו הוא הגיע לאיבר הבא שלה, אחרת אותם פעולות מהרשימה השנייה, וכאשר מגיעים לאיברים שווים מכניסים רק אחד מהם וקופצים יחד בשתי הרשימות כדי להימנע מאיברים כפולים. וכך עד לסיום המעבר בשני הרשימות.*

*זמן הריצה יהיה סכום גדלי 2 הרשימות הממוינות.*

*זמן ריצה:*

1. *הרעיון: שמירת מצביע לאיבר המינימלי, ובכל פעולה עדכון האיבר המינימלי כך שיהיה אפשר לשלוף אותו בקלות בשגרת minimum.*

*שגרת :*

*מעבר על כל הרשימה ובדיקה האם האיבר שרוצים להכניס קיים כבר ברשימה, במידה ולא, הכנסה לתחילת הרשימה.*

*בנוסף במידה והאיבר להכנסה מכיל ערך שקטן מהערך המינימלי הנוכחי של הרשימה, עדכון הערך המינימלי לערך שנכנס*

*(ישנה אפשרות גם להכניס לסוף הרשימה וכך לשמור ולתחזק מצביע לסוף הרשימה-לשימוש עתידי)*

*זמן ריצה:*

*שגרת :*

*החזרת הערך המינימלי השמור.*

*מעבר על איברי הרשימה עד לאיבר המינימלי ומחיקה שלו – (בעזרת מצביע לאיבר הקודם וההבא)*

*מעבר על \*כל\* איברי הרשימה תוך חיפוש המינימום החדש ושמירת מצביע לאיבר המינימלי ברשימה, ומצביע לאיבר שלפניו.*

*מחיקת האיבר המינימלי על ידי שינוי הצבעת האיבר שלפניו כך שהאיבר הבא יהיה האיבר שאחריו.*

*זמן ריצה:*

*שגרת :*

*החזרת האיבר המינימלי השמור*

*זמן ריצה:*

*שגרת :*

*מיון כל אחד מהרשימות בעזרת merge sort שמממומש בצורה דומה מאוד למקורי אך מותאם לרשימות מקושרות (עמיאל הראה מימוש פשוט באופק) ושימוש בUNION לרשימות מקושרות ממוינות מסעיף א' (שזמן הריצה שלו הוא ולא משפיע בסדר גודל על הזמן ריצה של השגרה) שעל הדרך יבטל כפילויות של איברים.*

*בנוסף עדכון המצביע לאיבר המינימלי החדש שיהיה המינימום של המינימליים של 2 הרשימות.*

*זמן ריצה: (בגלל המיון)*

*אוסיף הערה שגם לאחר קריאת הדיון הארוך בפורום עדיין כוונת התרגיל לא ברורה במאה אחוז (רק בהקשר של UNION – כלומר האם צריך למנוע כפילויות או לא)*

1. *השגרה היחידה שתהיה שונה היא השגרה UNION שבה לא נצטרך לבצע בדיקה שאין כפילויות ולכן הדבר היחיד שנצרך לעשות הוא חיבור 2 הרשימות, וזה מתבצע על ידי חיבור האיבר האחרון של אחת מהן לראשון של השני.*

*במידה ומתחזקים את הרשימה עם שמירת מצביע לסוף הרשימה זה יכול להיעשות ב אחרת יש צורך לעבור על כל איברי אחת מהרשימות וזמן הריצה יהיה*

**שאלה 5:**

1. נשתמש בנוסחה מהספר לשיטת החילוק:

1. *נשתמש בנוסחה מהספר לשיטת הכפל:*

1. *נחשב את האינדקסים לפי הסדר עבור מפתח k על פי האלגוריתם:*

*ניסיון 1 : עבור מחפשים ב*

*ניסיון 2: עבור מחפשים ב*

*ניסיון 3: עבור מחפשים ב*

*ניסיון 4: עבור מחפשים ב*

*ניסיון t: עבור מחפשים ב*

*אבל*

*וזו בדיוק הסכמה הריבועית עבור*

1. *נבצע את האלגוריתם מסעיף ג:*

*תחילה ולכן 63 יכנס לתא 63*

*כעת אבל תא זה תפוס ע"י 63 ולכן נוסיף 1 ונקבל*

*ולכן 2222 יכנס לתא 64*

*כעת אבל תא זה תפוס ע"י 63 ולכן נוסיף 1 ונקבל*

*אבל תא זה תפוס ע"י 2222 ולכן נוסיף 2 ונקבל*

*ולכן 190 יכנס לתא 66*

*נראה תוצאות דומות בעזרת משוואה 11.5:*

*תחילה עבור ולכן 63 יכנס לתא 63*

*כעת עבור אבל תא זה תפוס ע"י 63*

*ולכן עבור ו2222 יכנס לתא 64*

*כעת עבור אבל תא זה תפוס ע"י 63*

*ולכן עבור אבל תא זה תפוס ע"י 2222*

*ועבור ו190 יכנס לתא 66*

1. *תחילה ולכן 63 יכנס לתא 69*

*כעת ולכן 2222 יכנס לתא 24*

*כעת ולכן 190 יכנס לתא 44*

*נראה תוצאות דומות בעזרת משוואה 11.5:*

*תחילה עבור ולכן 63 יכנס לתא 69*

*כעת עבור ולכן 2222 יכנס לתא 24*

*כעת עבור ולכן 190 יכנס לתא 44*