



המכללה האקדמית להנדסה סמי שמעון

עבודת בית 3

מתרגלת אחראית: ויקטוריה בריימן

תאריך פרסום: 03.12.2020

תאריך הגשה: 17.12.2020 עד שעה 23.55

הנחיות כלליות:

- קראו היטב את השאלות.
 - ניתן להגיש את העבודה בזוגות.
 - הגשת העבודה תהיה דרך אתר הקורס במודל.
 - יש לכווץ כל קבצי עבודה לקובץ zip. (קובץ zip יכיל חלק תיאורטי בקבץ pdf וקבצי קוד C עבור חלק מעשי).
 - שם הקובץ שיוגש למערכת ההגשה יהיה מורכב מת"ז של המגיש/ים.
- לדוגמה:

עבור הגשה ביחיד - 111111111.zip

עבור הגשה בזוג - 111111111_22222222.zip

- אפשר להשתמש בקובץ השלד משיעור כבסיס לפתרון.
- במקרה של הגשה בזוגות, רק אחד מבני הזוג יגיש את העבודה במודל.
- חובה לתעד את הקוד.
- במידה ומצאתם טעויות בעבודה או בקשות להבהרות, אנא שלחו ל- victoria@ac.sce.ac.il תיקון/ הבהרה בהקדם.
- איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו. בכל מקרה לא יהיה ניתן להגיש מעבר ל-3 ימי איחור ממועד ההגשה המקורי. במקרים חריגים בלבד יש לפנות למרצה כדי לקבל אישור על הגשה באיחור.
- שאלות לגבי העבודה יש לשאול בפורום באתר הקורס ("מודל") או בשעות קבלה של המתרגל האחראי **בלבד**. אין לשלוח שאלות במייל המרצה.
- **להזכירכם: יש לשמור על הגינות אקדמית!**



המכללה האקדמית להנדסה סמי שמעון

חלק א' (תיאורטי) (40 נק.)

שאלה 1

השיטה Build-Max-Heap שלמדתם בכתה, יכולה להיות ממומשת גם בדרך אחרת – ע"י שימוש ב-Max-Heap-Insert. נתונה השיטה הבאה:

BUILD-MAX-HEAP-BY-INSERTION(A)

1. $heap-size[A] \leftarrow 1$
2. **for** $i \leftarrow 2$ **to** $length[A]$
3. **do** MAX-HEAP-INSERT(A, A[i])

- א. בהינתן מערך כלשהו A של מספרים, האם השיטות Build-Max-Heap(A) ו-Build-Max-Heap-By-Insertion(A) יבנו ערימות מקסימום זהות? אם כן הוכיחו, אחרת ספקו דוגמה נגדית.
- ב. הראו כי במקרה הגרוע Build-Max-Heap-By-Insertion תרוץ בזמן $O(n \log n)$ (בבניית ערימה בת n איברים).

שאלה 2

הוכיחו באינדוקציה שבערימה בת n איברים יש $\left\lceil \frac{n}{2} \right\rceil$ עלים.

שאלה 3

שרטטו עצי חיפוש בינאריים בגבהים 2, 3, 4, 5 עבור קבוצת המפתחות: $\langle 0, 2, 5, 8, 10, 15, 16 \rangle$

במידה ויש מספר אפשרויות עבור גובה מסוים, ציירו אחת מהן.

שאלה 4

כתבו גרסה רקורסיבית (פסאודו-קוד) של פונקציה TREE-INSERT (הכנסת איבר חדש לעץ חיפוש בינארי).

שאלה 5

בנו עץ חיפוש בינארי (BST) על ידי הכנסת הצמתים הבאים החל מעץ ריק (סדר הכנסת צמתים משמאל לימין):

25, 16, 10, 12, 8, 4, 9, 40, 15, 20, 3, 26, 50

שרטטו עץ אחרי מחיקת צומת 10



המכללה האקדמית להנדסה סמי שמעון

חלק ב' (מעשי) (60 נק')הקדמה:

משתמשים בעץ חיפוש בינארי כדי לשמור את ציוני התלמידים בקורס מסוים. בעץ זה, כל קדקוד מיצג תלמיד בקורס ומפתח המיון של העץ הוא תעודת הזהות של התלמיד.

המשימה:

- הגדירו מבנה (struct) בשם StudentNode. מבנה זה מגדיר צומת בעץ חיפוש בינארי המייצג סטודנט ומכיל את הפרטים הבאים עבור כל סטודנט:

- מספר תעודת הזהות
- ציון בוחן אמצע
- ציון מבחן

- הגדירו מבנה (struct) BST המייצג עץ חיפוש בינארי עבור תלמידי הקורס. מבנה זה יכול:
- מצביע לשורש העץ

- הוסיפו לתכנית את הפונקציות הבאות:

`void insert(BST *tree, int ID)`

פונקציה זו מכניסה תלמיד חדש בעל ת.ז. ID. אם תלמיד קיים, היא לא מבצעת דבר. שדות ציון בוחן אמצע וציון המבחן יוגדרו להיות 0.

`void updateMidtermGrade(BST *tree, int ID, int newGrade)`

פונקציה זו מעדכנת ציון בבוחן אמצע של תלמיד בעל מספר ת.ז. נתון. במידה ולא קיים תלמיד כנ"ל, לא מבצעת דבר.

`void updateExamGrade(BST *tree, int ID, int newGrade)`

פונקציה זו מעדכנת ציון מבחן של תלמיד בעל מספר ת.ז. נתון. במידה ולא קיים תלמיד כנ"ל, לא מבצעת דבר.

`int getFinalGrade(BST *tree, int ID)`

פונקציה זו מחזירה את ציון הסופי של תלמיד בעל מספר ת.ז. נתון. במידה ולא קיים תלמיד כנ"ל, מחזירה 0. הניחו כי המשקל של בוחן אמצע הוא 30% ממשקל הציון הסופי של תלמיד בקורס, ומשקל המבחן הוא 70% מציון זה. הציון הסופי יש לעגל. לדוגמה, אם



המכללה האקדמית להנדסה סמי שמעון

בחישוב יוצא ציון סופי 86.7, אחרי העיגול הציון הינו 87. אם בחישוב יוצא ציון 86.2, אז אחרי העיגול הציון הסופי הינו 86.

- `double getAverageGrade(BST *tree)`
פונקציה זו מחזירה את ממוצע בציונים הסופיים בקורס.
- `void reportGrades(BST* tree)`
פונקציה זו מדפיסה על המסך את פרטי התלמידים **ממוינים לפי ת.ז.** כל שורה נפרדת תייצג תלמיד ותכיל את פרטים הבאים (**משמאל לימין**):

ציון סופי ציון מבחן ציון בוחן אמצע מספר ת.ז.

הפרטים בכל שורה צריכים להיות מופרדים ע"י tab.

- `int getNumOfFails(BST *tree)`
פונקציה זו מחזירה את מספר הנכשלים בקורס. ציון נכשל הינו ציון מתחת ל-56.
- `void ReportStatistics(BST* tree)`
פונקציה זו מדפיסה על המסך את מספר תלמידים בשלושת דרגת ציון המיוצגת סטטיסטיקה ציונים בקורס:

	Degree	final grade
Good	(56 – 75)	
Very good	(76 – 94)	
Excellent	(95 -100)	

- מותר ומומלץ להוסיף פונקציות עזר
- אין לשנות את החתימות של הפונקציות!

צב/דה נצימה!