



המכללה האקדמית להנדסה סמי שמעון

עבודת בית 2 - מעשי

מתרגלת אחראית: ויקטוריה בריימן

תאריך פרסום: 19.11.2020

תאריך הגשה: 03.12.2020 עד שעה 23.55

הנחיות כלליות:

- קראו היטב את השאלות.
 - ניתן להגיש את העבודה בזוגות.
 - הגשת העבודה תהיה דרך אתר הקורס במודל.
 - יש לכתוב תכנית המכילה שלושה קבצים (קובץ הגדרות - קובץ עם סיומת h, קובץ המכיל כל גופי פונקציות וקובץ ראשי - main) בשפת תכנות C. קבצי תכנית – הראשי והמכיל כל פונקציות יכולים לקבל סיומת cpp, אבל חייבים להיכתב בחוקים של שפת תכנות C.
 - יש לכווץ כל קבצי התוכנית לקובץ zip.
 - שם הקובץ שיוגש למערכת ההגשה יהיה מורכב מת"ז של המגיש/ים.
- לדוגמה:

עבור הגשה ביחיד - 11111111.zip

עבור הגשה בזוג - 11111111_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 המתרגלות או המרצה.
- להזכירכם: יש לשמור על הגינות אקדמית!



המכללה האקדמית להנדסה סמי שמעון

מטרות התרגיל:

- שימוש במבנה נתונים: עץ בינארי.
- לפני שמתחילים יש לחזור על חומר הנלמד בנושא עצים בינאריים.

כללי:

- עליהם לכתוב פונקציות הבאות המטפלות בעץ בינארי:

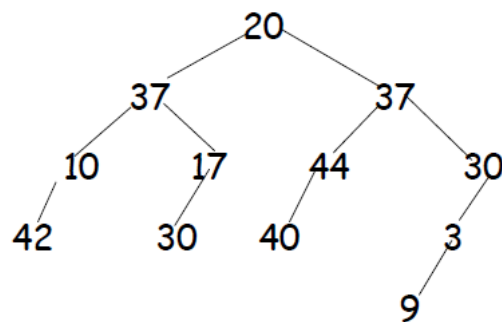
1. כתבו פונקציה `int isEmpty(node* root)` המקבלת מצביע לשורש של עץ בינארי, ובודקת האם העץ הוא ריק.

הפונקציה מחזירה ערך 1 אם עץ ריק, ו-0 במקרה אחרת.

2. כתבו פונקציה `void printLeaves(node* root)` המדפיסה את כל עלים בעץ.

יש להדפיס את העלים משמאל לימין עם רווח יחיד ביניהם.

למשל, עבור עץ שבציר הפונקציה תדפיס (משמאל לימין): 42 30 40 9.



3. כתבו פונקציה `int contains(node* root, int key)` הבודקת האם מפתח `key` נמצא בעץ בינארי.

- אם `key` מופיע בעץ, הפונקציה תחזיר ערך 1
 - אם `key` לא מופיע בעץ – הפונקציה תחזיר 0.
- למשל, עבור עץ מסעיף הקודם:

- עבור ערך חיפוש 44 הפונקציה תחזיר 1
- עבור ערך 12 הפונקציה תחזיר 0.



4. כתבו פונקציה `int countNodes(node* root)` לחישוב כמות הכוללת של צמתים בעץ.

לדוגמה: עבור העץ משאלה 2 הפונקציה מחזירה 12.

5. כתבו פונקציה `int sumOfKeys(node* root)` המחשבת ומחזירה סכום ערכי הצמתים

של עץ בינארי.

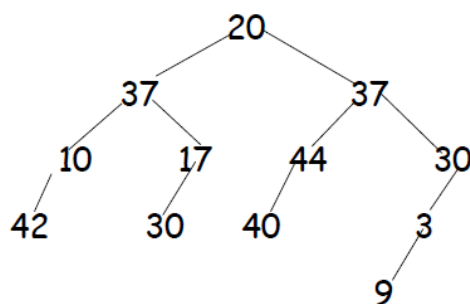
לדוגמה: עבור העץ משאלה 2 או עבור העץ משאלה 6 (בשתי שאלות הוצג אותו עץ)

הפונקציה מחזירה 319.

6. כתבו פונקציה `void levelStatistics(node* root, int level)` המדפיסה מפתחות של

כל הצמתים ששייכים לרמה `level` וגם מחשבת ומחזירה את מספר הצמתים ברמה.

לדוגמה עבור העץ שמשאל:



• לרמה 0 – הפונקציה מדפיסה 20

ומספר צמתים 1

• לרמה 2 – הפונקציה מדפיסה

מפתחות (משמאל לימין):

30 44 17 10 ומספר צמתים 4

תחילה יש להדפיס את המפתחות משמאל לימין עם רווח יחיד ביניהם

לאחר מכן, בשורה הבאה יש להדפיס את מספר הצמתים ברמה זו.

7. כתבו פונקציה `int isPerfect(node* root)` הבודקת האם עץ הוא עץ מושלם (perfect).

אם העץ משולם, הפונקציה תחזיר 1, אחרת היא תחזיר 0.

צהודה נצימה!