

# מכללה אקדמית הדסה

## החוג למדעי המחשב

### תרגיל #13 בקורס: תכנות מודולארי

#### עצים בינאריים

בתרגיל זה יש שלוש שאלות. עליכם לתכנת את תשובותיכם בקבצים ששםם הוא `ex13a.cc`, `ex13b.cc`, `ex13c.cc`.

**תכנית a: מספר הערכים החיוביים, שליליים, אפסיים בעץ (33%).**  
כתבו תכנית הקוראת סדרת מספרים שלמים, ובונה מהם עת חיפוש בינארי. יש לקרוא את הקלט עד `.eof`.

ענה התכנית מזמנת פונ' אשר סופרת ומחזירה: צמתים בעץ מקיימים שהערך בצומת, והערך בבנו השמאלי שווים

התכנית הראשית מציגה את הערך שהפונ' החזירה.

#### **תכנית b: תת עץ שכל הנקודות בו מצויות באותו רביע (33%).**

הגדירו בתכנית `struct Point` כפי שאתם מכירים, וכן:

```
struct Node {  
    int _id ;  
    struct Point * _data ;  
    struct Node * _left, * _right ;  
};
```

בנו עץ חיפוש בינארי ממיון לפי המרכיב `_id` ('המזהה' של כל נקודה).  
בנו את העץ ע"י קריאת סדרת שלשות (מזהה כל נ', ערך x שלה, ערך y שלה) עד `.eof`. הניחו כי הקלט תקין.

זמנו פונ' המקבלת מצביע לשורשו של העץ הנ"ל ומחזירה מצביע לתת העץ הגדול ביותר המקיים שכל הנקודות שבו מצויות באותו רביע.

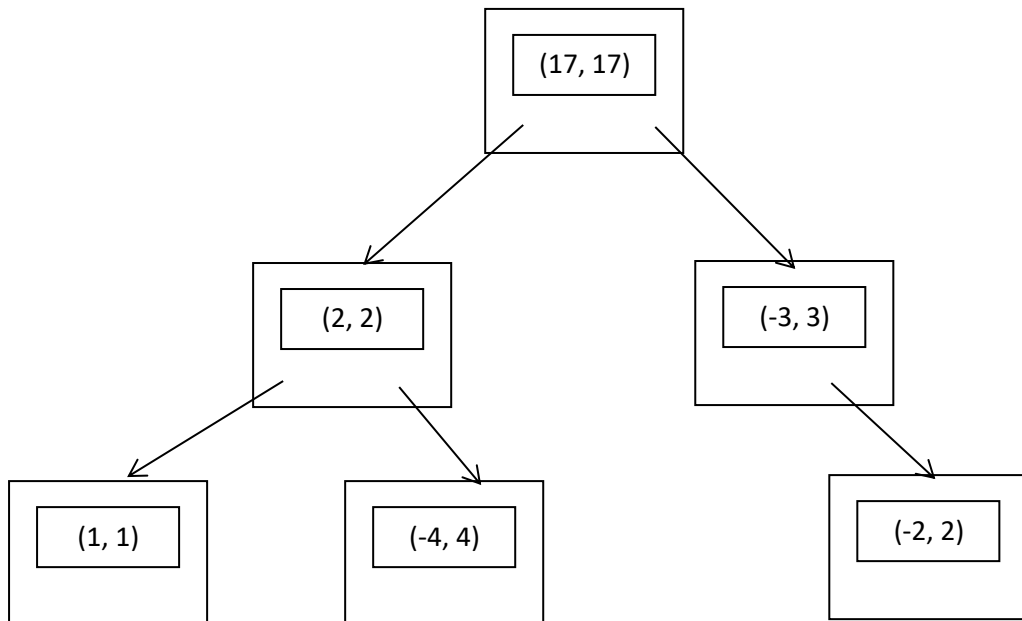
בתכנית הראשית הציגו את הנקודה שמצויה בצומת שמצביע לו הוחזר (או את הזוג 0 0 אם אין נקודה כנדרש).

#### הערות:

- הקפידו על היעילות!
- ראוי שהפונ' שתכתבו תקבל ותחזיר בדיוק את הערכים המתוארים בשאלה; הפונ' רשאית, כמובן, לזמן כל פונ' אחרת שתראו לנכון.
- במידה ובעץ קיימים כמה תתי-עצים מקסימאליים בגודלם החזירו את הראשון אליו מגיעים בביקור תוכי (`inorder`).
- ניתן להניח שאין בעץ נקודות המצויות על-גבי הצירים.

ה. במידה והעץ ריק יש להחזיר NULL. במצב זה הפלט יהיה 0 0

לדוגמה: בעץ הבא:  
(מהציר הושמטו מצביעים שערכם NULL)



יוחזר מצביע לצומת של  $(-3, 3)$  שכן שני הצמתים בתת-עץ זה מצויים באותו רביע, ואין תת-עץ גדול יותר שכל הצמתים בו מצויים באותו רביע.

לו בצומת המכיל את הנקודה  $(-4, 4)$  הייתה במקום נקודה זאת הנקודה:  $(4, 4)$  אזי היה מוחזר מצביע לתת-העץ של  $(2, 2)$  שכן תת-עץ זה היה מקיים שכל הנקודות בו מצויות באותו רביע, והוא גדול יותר מתת-העץ של  $(-3, 3)$ .

הסבירו בתיעוד הראשי מהו זמן הריצה של הפונ' שכתבתם.

## תכנית c: תת העץ המקסימלי עם רוב זוגי (34%).

כתבו תכנית המגדירה::

```
struct Node {  
    int _data ;  
    struct Node *_left, *_right;  
} ;
```

התכנית תכיל פונקציה ראשית שתקרא סדרת מספרים שלמים, ותבנה מהם עץ חיפוש בינארי כפי שלמדנו בכיתה. יש לקרוא את הקלט עד eof.

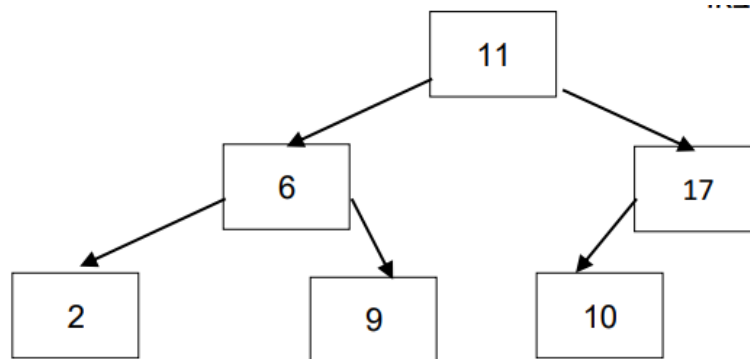
כתבו פונ' אשר מקבלת מצביע לשורש העץ, ומחזירה מצביע לתת העץ הגדול ביותר שמקיים שכמות הערכים הזוגיים בו גדולה ממש מכמות הערכים האי-זוגיים בו.

הפונקציה הראשית תקרא לפונקציה ותדפיס:

באם נמצא תת עץ שמקיים את התנאי, היא תדפיס את הערך המצוי בתת העץ שחזר מהפונקציה.

אם לא נמצא תת עץ כנדרש יש לכתוב לפלט "NOT FOUND" - בדיוק כך.

לדוגמה: עבור העץ הבא:



יוחזר מצביע לצומת של 6 (בתת עץ זה יש שני ערכים זוגיים, וערך אחד אי זוגי). התכנית תדפיס את הערך 6.

לו הערך בצומת של 17 היה זוגי כלשהו (ולא 17) אזי היה מוחזר מצביע לצומת של 11 (שכן אז בעץ השלם היו ארבעה ערכים זוגיים ושניים פרדיים) והיה מודפס 11.

הערות:

- ראוי שהפונ' שתכתבו תקבל פרמטרים בדיוק כפי שמתואר בשאלה. הפונ' רשאית לזמן כל פונ' שהיא, ולהעביר לה פרמטרים כפי שיקול דעתכם.
- הקפידו על היעילות.