משימות

1. חלק יבש
   1. הסבר ותיאור מבני הנתונים והאלגוריתמים בהם השתמשנו
      1. שווה להשתמש בציור
   2. הוכחת סיבוכיות הזמן ומקום
   3. **לא יותר מ-8 עמודים!**
2. לחלק עבודה (פונקציות ומבנים)
3. טסטים
   1. טסטים רשמיים
   2. טסטים מהפייסבוק
   3. טסטים בשרת
4. לתקן קוד (לפי באגים ולפי קונבנציות)
5. להשוות דרישות במסמך למה שעשינו
   1. לעבור על השאלות בפורום
6. ~~לבנות גיט~~

**לזכור-**

* דליפות זיכרון
* ולגרינד

חלוקת עבודה

|  |  |
| --- | --- |
| **אביטל** | **אלון** |
| מחלקת Generic\_AVL | לכתוב יבש וארכיטקטורה |
|  | מחלקת Artist |
|  |  |
|  |  |

**בהמשך:**

* כתיבת עצי AVL יורשים
* חלוקת פונקציות למימוש

שאלות

1. ב-DRY1?

רעיון כללי למימוש

**צריך:**

* מחלקה גנרית של עץ AVL
  + תומכת בכל הפעולות שצריך לעשות בעץ-
    - הוספת צומת
    - הוצאת צומת
    - חיפוש צומת
* מחלקה של אמן
  + ARTISTID
  + מערך של כלל השירים (מערך של מצביעים?)
  + עץ AVL של מספר ההשמעות שיש לכלל השירים
* מחלקה של שיר?
  + כנראה שמספיק להחזיק פוינטר?

**המבנה:**

* עץ AVL כללי של אמנים
  + כל צומת - ARTIST
* עצי AVL של מספר ההשמעות
  + כל צומת – עץ AVL של שירים
* עצי AVL של שירים
  + כל צומת – SONGID
* רשימה מקושרת של "מספר השמעות"
  + עץ AVL של אמנים (שיש להם את אותו מספר השמעות לפי האיבר)

\* כשמוחקים אמן – אכן mlogn

* למה – כי אנחנו רצים על המערך של השירים של אותו אמן, מוצאים לפי הסדר בעץ ה-AVL של האמנים ולמחוק את כל השירים שם – זה מסתכם ל-mlogn+m (צריך להסביר למה)
* להסביר שאר ענייני סיבוכיות

קונבנציית קוד

לסדר קוד שיהיה יפה ולוודא קונבנציות:

1. הערות
   1. דברים לא מובנים
   2. תיעוד
2. Refactoring
   1. שכפול קוד
   2. לפרק לפונקציות
3. קונבנציית קורס?
   1. אורך שורות
   2. אורך פונקציות
   3. קבועים בקוד
   4. פונקציות מתחילות באות קטנה
   5. לשנות שמות משתנים למובנים + קונבנציה נכונה
   6. לבדוק הזחות – טאב (4 רווחים, אבל כטאב ולא רווחים)

שונות

* + - 1. לוודא קימפול על השרת לאורך הדרך
      2. לבנות מחלקה ב-cpp?
      3. אסור להשתמש ב-STL
      4. להבין את ה-main