

## תרגיל בית מספר 4

**נושא: עץ חיפוש בינארי ולהק רחפנים פולימורפי**

**דדליין: יום ד', 22/01/2025, 23:59**

**הגשה ביחידים**

**בהצלחה רבה!**

### תיאור התרגיל

תרגיל זה מהווה הרחבה של תרגיל בית 3 בו מימשתם עולם מונחה עצמים של רחפנים המחפשים נעדר בתוך יער. בתרגיל הנוכחי יתווספו שני מרכיבים לסימולציה הקיימת – ראשית מבנה נתונים שיושמש עבור הרחפנים ושנית, טיפוס רחפנים נוספים והתנהגות פולימורפית של הרחפנים.

### חלק א': עץ חיפוש בינארי.

"עץ חיפוש בינארי" הינו מבנה נתונים סדור לפי העיקרון הרקורסיבי "איברי תת עץ שמאלי קטנים בערכם מן השורש, איברי תת-עץ ימני גדולים בערכם מן השורש". ההנחה היא כי יחס סדר מוגדר היטב על הערכים של האיברים המיועדים לעץ.

**המימוש שלכם נדרש לכלול את הרכיבים הבאים:**

- פונקציה בשם `insert` המכניסה לעץ איבר באופן השומר על העיקרון הרקורסיבי הנ"ל, ההכנסה תבוצע תחת הנחה שאופרטור `<` ממומש עבור טיפוס הנתונים. לא תיתכנה כפילויות בעץ – הכנסת איבר שכבר קיים בעץ תגרום לדריסה.
- פונקציה בשם `search` המחפשת איבר בעץ ע"פ ערכו ומחזירה מצביע אליו אם וכאשר נמצא.
- פונקציה בשם `min` המחזירה מצביע אל האיבר בעל הערך המינימלי בעץ
- פונקציה בשם `max` המחזירה מצביע אל האיבר בעל הערך המקסימלי בעץ
- פונקציה בשם `remove` המוחקת איבר נתון מן העץ בהנחה שהינו קיים בעץ
- פונקציה בשם `clear` המנקה את העץ (מוחקת את כל האיברים בעץ)
- פונקציה בשם `print` המדפיסה את כל איברי העץ באופן סדור. סדר הדפסת האיברים הינו סדר המיון העולה. הפונקציה תסתמך על אופרטור הדפסה לפלט `<<` של האיבר.
- פונקציה בשם `size` המחזירה את מספר האיברים בעץ.

מעבר לאפיון ולדרישות אלו, אתם חופשיים לתכן ולממש כרצונכם מבנה נתונים זה, כל עוד נשמרת הפונקציונליות המתוארת. שימו לב שאתם רשאים לדרוש תכונות מסוימות שעל האיברים לקיים כדי שיאוחסנו במבנה הנתונים שלכם (בדומה לאופרטור שהוזכר בפונקציית ההכנסה). **עליכם לפרט בתיעוד את כל הדרישות וההנחות להן מבנה הנתונים נזקק.**

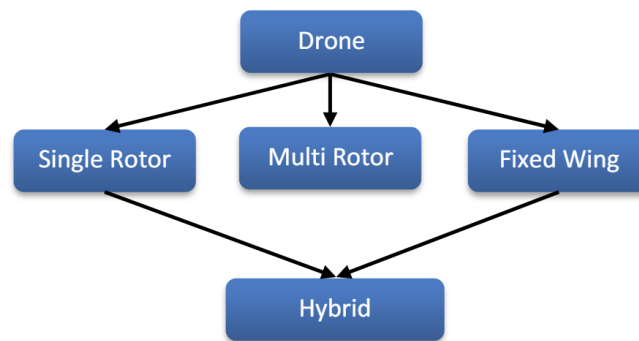
## חלק א': עץ חיפוש בינארי.

בחלק זה של התרגיל אתם תשדרגו את הסימולציה אותה מימשתם בתרגיל הקודם ללהק בעל רחפנים פולימורפיים הממלאים תפקידים שונים.

מעתה, סוגי הרחפנים בהם הסימולציה תתחשב הינם:

- Single Rotor Drone – רחפן בעל רוטור (מדחף) יחיד ('S')
- Multi Rotor Drone – רחפן בעל מספר רוטורים ('M')
- Fixed Wing Drone – רחפן בעל כנפיים ('W')
- Fixed Wing Hybrid VTOL – רחפן שהוא שילוב של Fixed Wing ו-Single Rotor ('H')

עץ הירושה המתאר את להק הרחפנים נתון בתרשים הבא:



עדכון הרחפנים ימשיך להתרחש בכל פעימת זמן בדידה (איטרציה), וישמור על אופיו ההסתברותי, אך העדכון עצמו יהיה תלוי בסוג הרחפן – כפי שיוסבר כאן בפירוט. תנועה של כל רחפן ביער תתרחש לפי משוואת התנועה המקורית, כלומר עבור רחפן בעל אינדקס  $i$  מיקומו בנקודת זמן  $t + 1$  ייקבע ע"פ המשוואה הבאה:

$$\vec{l}_i(t + 1) = \vec{l}_i(t) + \vec{v}_i(t)$$

ההבדל בין הרחפנים השונים יבוא לידי ביטוי בעדכון רכיב המהירות. מהירות של כל רחפן תמשיך להיות תלויה, כמו בתרגיל הקודם, בשלושה מרכיבים – בערך הקודם, במוביל הקבוצה ( $global\ best$ ),  $\hat{y}(t)$  ובשיא האישי ( $personal\ best$ ),  $\vec{y}_i(t)$ :

$$\vec{v}_i(t + 1) = \alpha \cdot \vec{v}_i(t) + \beta \cdot r_{1i}(t) \cdot (\vec{y}_i(t) - \vec{l}_i(t)) + \gamma \cdot r_{2i}(t) \cdot (\hat{y} - \vec{l}_i(t))$$

הערכים של הפרמטרים  $\{\alpha, \beta, \gamma\}$  ייקבעו ע"פ סוג הרחפן:

- לרחפן עם רוטור אחד ('S') יהיו פרמטרים בעל ערכים  $\{\alpha_S = 0.01, \beta_S = -0.1, \gamma_S = -0.25\}$
- לרחפן עם מספר רוטורים ('M') יהיו פרמטרים בעל ערכים  $\{\alpha_M = 0.05, \beta_M = 0.1, \gamma_M = 0\}$
- לרחפן בעל כנפיים ('W') יהיו פרמטרים בעל ערכים  $\{\alpha_W = 0.25, \beta_W = 1, \gamma_W = 1\}$
- רחפן הייברידי ('H') ינוע כמו רחפן עם רוטור אחד.

# תכנות בשפת ++C, סתיו 2024-5

## דרישות מימוש

- בכדי להשמיש את מבנה הנתונים, ממשו אופרטור הדפסה לערוץ הפלט הסטנדרטי << אשר מדפיס את טיפוס הרחפן ואת מיקומו הנוכחי.
- התכנית תקבל גם בתרגיל זה שני קבצי קלט – קובץ קונפיגורציה וקובץ אתחול, אשר כוללים שינוי בפורמט שלהם. **קובץ הקונפיגורציה** - יכיל את השורות הבאות לפי הסדר:
  - שורה המכילה ארבעה ערכים שלמים,  $x_{min}$ ,  $y_{min}$ ,  $x_{max}$ ,  $y_{max}$  המגדירים את שטח היער והקצוות שלו.
  - שורה המכילה 2 ערכים ממשיים עבור מיקום הנעדר (נקודת המטרה), כמו בתרגיל הקודם.
  - מספר שלם חיובי המייצג את מספר האיטרציות המקסימלי, כמו בתרגיל הקודםלהלן קובץ קלט חוקי עבור קונפיגורציה של הסימולציה, (דוגמא זו מתארת את הגדרת השטח עבור התרגיל הקודם):

```
0 0 72 42
10.5 2.5
1000000
```

יש לוודא שאכן  $x_{min} < x_{max}$  ו-  $y_{min} < y_{max}$ , ושאכן נקודת המטרה נמצאת בתוך התחום שניתן.

**קובץ האתחול** – בדומה לתרגיל הקודם יכיל את מספר הרחפנים ומיקומם בשטח היער **בתוספת סוג הרחפן**. הקובץ יכיל בשורה הראשונה את מספר הרחפנים בקבוצה, ובשורות לאחר מכן את סוגם ומצבם ההתחלתי.

כל שורה תתחיל בתו המגדיר את טיפוס הרחפן (התווים האפשריים הם  $\{ 'S', 'M', 'W', 'H' \}$ ), ולאחר מכן תכיל את מיקומו ההתחלתי של הרחפן (באמצעות 2 קואורדינטות) ואת מהירותו ההתחלתית של אותו הרחפן (באמצעות 2 רכיבי מהירות), מופרדים ברווחים. להלן דוגמא עבור קובץ קלט חוקי של אתחול הרחפנים:

```
5
S 0 0 0.5 0.5
M 0.1 0.1 0.2 0.2
W 0 0.1 0 0.3
H 0.1 0 0.3 0
M 1.33 1.66 0.6 0.6
```

- **פלט התכנית** - בסיום מעברי הזמן, על התכנית להחזיר כפלט את מספר צעדי הזמן של הסימולציה וכן את מיקומי הרחפנים של צעד הזמן האחרון באמצעות כתיבה לקובץ **ששמו התקבל כארגומנט לתכנית כמו בתרגיל הקודם**. כאשר סדר הרחפנים יהיה כפי שניתן בקובץ הקלט (הרחפן הראשון בקובץ הקלט יהיה הראשון בפלט וכו'). למשל כך יכול להיראות הפלט עבור הדוגמא שלעיל (זוהי רק דוגמא):

```
96717
S 3.01 13.96
M 10.45 7.51
W 13.87 20.9
H 58.18 14.84
M 57.99 17.1
```

- **גרעין אקראיות** - לעומת התרגיל הקודם, התכנית תוכל לקבל ארגומנט רשות נוסף דרך שורת הפקודה שיהיה מספר שלם שיהווה את הגרעין (seed) עבור הפונקציה **srand**. במידה וארגומנט זה נמסר והוא אכן מספר שלם, יש להפעיל את הפונקציה **srand** בעזרת המספר שהתקבל. במידה וארגומנט זה לא נמסר, או שאינו מספר שלם, יש לקרוא ל **srand** באמצעות הזמן הנוכחי המיוצג באמצעות הזמן הנוכחי (**srand(time(NULL))**). להלן דוגמאת הרצה של התכנית לאחר קומפילציה של הקוד לכדי קובץ הרצה בשם **droneRescue** יחד עם ארגומנט הרשות הנוסף:

```
$ ./droneRescue config.dat init.dat final.dat 123456
```

# תכנות בשפת ++C, סתיו 2024-5

## הנחות עבודה

- הנחות העבודה מהתרגיל הקודם (מספר מעברי זמן, כמות רחפנים) תקפות גם בתרגיל זה.
- ניתן להניח שתחומי היער בקובץ הקונפיגורציה הם חיוביים.
- בהינתן מיקום של רחפן בקובץ הקלט, שנמצא מחוץ לשטח היער, יש לקבע אותו לתחום בהתאם לתנאי השפה.
- הטיפול נותר כמו בתרגיל הקודם – במקרה של קלט לא חוקי (למשל, שגיאת קבצים, חוסר עקביות במימדי היער וכיוצא באלה) התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאה הסטנדרטי:

Error; invalid input

## דגשים

- עליכם לעשות שימוש במבנה הנתונים אותו מימשתם בחלק א' של התרגיל בכדי לאחסן את הרחפנים באופן פולימורפי; כלומר, אובייקט היער מצופה להחזיק את הרחפנים באמצעות מבנה זה. מיון הרחפנים בעץ יהיה על-סמך המספר הסידורי שלהם, אשר נקבע ע"פ סדר הצגתם בקובץ הקלט בזמן אתחול הסימולציה.
- הקדישו מחשבה למבני הנתונים השונים בהם תעשו שימוש; הביאו בחשבון שיקולי יעילות.
- יש לבדוק תקינות קלטים ולהציג הודעות שגיאה מתאימות.
- בתרגיל בית זה אין להשתמש בספריית STL; מבני הנתונים והאלגוריתמים צריכים להיות ממומשים על-ידיכם.
- עליכם לוודא כי התכנית עוברת קומפילציית ++g התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות קצרות, אך משמעותיות, המתארות את הפונקציות השונות.
- יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה: hwcheck

## הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('63'), שם התרגיל ('ex4') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא: 63\_ex4\_studID.zip.
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות \*.h \*.cpp \*.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם 63\_ex4\_012345678.zip הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.  
לא תתקבלנה הגשות באיחור!