



# מבני נתונים 234218 חורף תשפ"ז

גילון רטוב מס' 2 – מעודכן לתאריך 04.01.2026

עמוד 1 מתוך 9

מתרגל ממונה על התרגיל: יצחק גרינבוים, [Yitzchakg@campus.technion.ac.il](mailto:Yitzchakg@campus.technion.ac.il)

תאריך ושעה הגשה: 21/01/2026 בשעה 23:59

בזוגות. אין להגיש ביחידים. (אלא באישור מתרגל אחראי של הקורס)

אופן ההגשה:

## הנחיות כלליות:

- שאלות על התרגיל יש לפרטם באתר הפיאצה של הקורס תחת לשונית "hw-wet-2":
  - האתר: <https://piazza.com/technion.ac.il/winter2026/234218>, נא לקרוא את השאלות של סטודנטים אחרים לפני שמספרם שאלת חדשה, למקורה שנשלה כבר.
  - נא לקרוא את המסמך "נהלי הקורס" באתר הקורס. בנוסף, נא לקרוא בעיון את כל ההנחיות בסוף מסמך זה.
  - בפורום הפיאצה ינוהל FAQ ובמידת הצורך יועלו תיקונים כהודעות נועצות (Pinned Notes). תיקונים אלו חשובים.
  - התרגיל מורכב משני חלקים: יש ורטוב.
  - לאחר קיראת כל הדרישות, מומלץ לתכנן תחילת את מבני הנתונים על נייר. דבר זה יכול לחסוך לכם זמן רב.
  - לפני שתתמודד באתגר, ודאו כי יש לכם פתרון העומד בכל דרישות הסביבות בתרגיל.
  - תרגיל שאינו עומד בדרישות הסביבות ייחסב כפסול.
  - את הפתרון שלכם מומלץ לחלק למחוקות שונות שאפשר למש (ולבדוק!) בהדרגות.
  - המלצות לפתרון התרגיל נמצאות באתר הקורס תחת: "Programming Tips Session".
  - המלצות לתכנות במסמך זה אינן מחייבות, אך מומלץ להיעזר בהן.
  - חומר התרגיל הינו כל החומר שנלמד בהרצאות ובתרגילים עד מינימום (לא כולל).
  - העתקת התרגיל בית רטוביים תיבדק באמצעות תוכנת בדיקות אוטומטית, המזהה דמיון בין כל העבודות הקיימות במערכת, גם אלו משנים קודמות. לא ניתן לערער על החלטת התוכנה. התוכנה אינה מבדילה בין מקור להעתיק! أنا הימנע מהסתכלות בקוד שאינו שלי.
  - בקשה להגשה מאוחרת יש להפנות למתרגל האחראי בלבד: [goldshtein@campus.technion.ac.il](mailto:goldshtein@campus.technion.ac.il).
  - בהצלחה!



## הקדמה:

"יצורים נדרים.  
מפלצות אימתניות.  
אוצרות יקר ערך.  
מלוכות שטניות.  
עלמות נסתרים.

הצירוף 'לא-נודע' מכל מעין קסם משלו. ישנו אלה שנשבו בקסם זהה.

קוראים להם **האנטרים** (צדדים)



האנטרים הם ציידי-עלית בעלי רישיון מטעם אגדת ההאנטרים. יכולותיהם הפיזיות, המנטליות ובוקר שליטם בון – אנרגיית חיים הניננת לתיעול לחיזוק הגוף, מניפולציה של חפצים, חישה מרחוק ויכולת התקפיות מיוחדות, מאפשרות להם לבצע שימוש שלא כל אדם מסוגל להן, כמו חקר אזורים מסוכנים, ציד יצורים נדרים, לוווי, הגנה ופתרון סכוכים מסוכנים. ולעתם, לא כל האנטר הוא גיבור. נן יכול לשמש לטוב או לרע. יש הפעלים למען האנשיות, ויש הרודפים אחר אינטרסים אישיים, עשר או כוח, ולעתים עד כדי עימותים עקובים גם בין האנטרים

שונים. לעיתים קרובות האנטרים מתאגדים לצוותים שפועלים יחד. יושב ראש ה-12 של הארגון, אייזק נטרו, שמע על יכולותיכם כמתכנת עילית ולא מכיר את chatgpt כל כך טוב, מעוניין להבהיר את המערכת הישנה למעקב אחר פעולות האנטרים למערכת חכמה חדשה בשם 'HunTech', והוא מבקש את עזרתכם בפיתוחה.

כל האנטר או צוות במערכת יהיו מיוצגים על ידי מספר מזהה ייחודי.



## תפקידים והנחות של מערכת ההאנטרים

- האנטר לא עוזב צוות שהוא פועל בו.
- כשותם מאבד את חבריו בקרב, הצוות הופך ללא פועל, אבל המידע על האנטרים נשאר במערכת.
- האנטרים בצוות מסוימים לפי הסדר הכרונולוגי בו הטרפו.
- לכל האנטר יש **יכולת נן** (NenAbility), שהיא אחת מ-6 אפשרויות: משפר (Enhancer), משנה (Transmuter), מכשף (Conjurer), פולט (Emitter), מתמן (Manipulator), ומומחה (Specialist). ובנוסיף **הילה** (Aura), שלם אי שלילי המתאר את כמות הנן שיש לאנטר.
- לכל צוות **יכולת נן משותפת**, שהיא סכום יכולות הנן של חברי (הסביר בנקודות למטה), וגם **ניסיונו** (מספר שלם).
- צוות האנטרים יכול לכפות על צוות אחר להctrוף אליו במידה ותנאים מסוימים מתקיימים (מפורט היכן שרלוונטי).
- במקרה כזה, כל האנטרים שהו בצוות הכהפה ייחסבו קודמים בסידור הכרונולוגי של ההctrוף לצוות. במפורש, אם A כופה את B להctrוף אליו:

$$A = (a_1, \dots, a_{|A|}), \quad B = (b_1, \dots, b_{|B|}) \quad \Rightarrow \quad A \text{ forces } B \text{ yields } (a_1, \dots, a_{|A|}, b_1, \dots, b_{|B|})$$

## נקודות חשובות

- סיפקנו לכם את המחלקה NenAbility, עם פעולות יצרה, העתקה, השמה, בדיקת תקיןות, איבר אפס, פעולות אРИתמטיות (=,-,+,-,+), שלילה, אופרטור השוואה (>,<,==), יכולת נן אפקטיבית והדפסה. אין צורך למשש את המחלקה זו בעצמכם.
- בכל אחת מהפונקציות שזמן הריצה שלה משוערך פרט לפונקציה `add_hunter`, השעריך הוא יחד עם כל הפונקציות האחרות פרט לפונקציה `add_hunter`.

**דרוש מבנה נתונים למימוש הפעולות הבאות:****Huntech()**

מאתחלת מבנה נתונים ריק. תחילת אין במערכת צוותים או האנטרים.

פרמטרים: אין.ערך החזרה: אין.סיבוכיות זמן: O(1) במקורה הגרוע.**virtual ~Huntech ()**

הפעולה משחררת את המבנה (כל הזיכרון אותו הקצתם חייב להיות משוחרר).

פרמטרים: אין.ערך החזרה: אין.סיבוכיות זמן: O(n + k) במקורה הגרוע, כאשר n הוא מספר האנטרים הכלול במערכת ו k הוא מספר הצוותים מאז תחילת המערכת.**StatusType add\_squad(int squadId)**הצotta בעל המזהה `squadId` נרשם באגודות האנטרים, אך יש לצרפו למערכת. צוות אין ניסיין בכלל. בעת ההכנסה אין האנטרים בצוות.פרמטרים:

מזהה הצוות החדש.

squadId

ערך החזרה:

ALLOCATION\_ERROR

INVALID\_INPUT

FAILURE

SUCCESS

במקרה של בעיה בהקצאה/שחרור זיכרון.  
אם `squadId` ≤ 0. סיבוכיות זמן: O(log k) במקורה הגרוע, כאשר k הוא מספר הצוותים במערכת.**StatusType remove\_squad(int squadId)**חברי הצוות בעל המזהה `squadId` נהרגו בקרב איום, וכן צריך להוציאו מהמערכת, יחד עם כל חברי. לאחר המלחמה, יתכן שייתווסף למבנה צוות אחר בעל אותו מזהה.פרמטרים:

מזהה הצוות.

squadId

ערך החזרה:

ALLOCATION\_ERROR

INVALID\_INPUT

FAILURE

SUCCESS

במקרה של בעיה בהקצאה/שחרור זיכרון.  
אם `squadId` ≤ 0. סיבוכיות זמן: O(log k) במקורה הגרוע, כאשר k הוא מספר הצוותים במערכת.**StatusType add\_hunter (int hunterId, int squadId, const NenAbility &nenAbility, int aura, int fightsHad):**האנטר בעל המזהה `hunterId` מצטרף לצוות בעל המזהה `squadId`. יכולת הנן שלו היא `nenAbility`, וההילה שלו היא `aura`. בנוסף שוהאנטר השתתף במספר `fightsHad` קרבנות.

# מבני נתונים 234218 חורף תשפ"ז

גילון רטוב מס' 2 – מעודכן לתאריך 04.01.2026

עמוד 4 מתוך 9



## פרמטרים:

מזהה האנטר שציריך להוסיף.

מזהה הוצאות של ההאנטר.

אובייקט שמתאר את יכולת הנן של ההאנטר.

מספר שלם המתאר את כמות הנן של ההאנטר.

מספר הקרבות בהם השתתף ההאנטר.

hunterId

squadId

nenAbility

aura

fightsHad

## ערך החזרה:

ALLOCATION\_ERROR

INVALID\_INPUT

FAILURE

SUCCESS

במקרה של בעיה בהקצאה/שחרור זיכרון.  
(wet2util.h) אם  $0 \leq \text{squadId} \leq \text{hunterId}$  או אם  $\text{nenAbility}$  לא תקין

.  
או  $< 0$  או  $> 0$  aura

אם קיימים כבר האנטר עם מזהה Id<sub>1</sub>, hunterId, או שהוצאה עם המזהה Id<sub>2</sub> squadId לא קיימים במערכת.  
עם מזהה Id<sub>1</sub>, hunterId, או שהוצאה עם המזהה Id<sub>2</sub> squadId לא קיימים במערכת.

במקרה של הצלחה.

סיבוכיות זמן:  $O(\log k)$  משוערך, ממוצע על הקלט, כאשר  $k$  הוא מספר הוצאות במערכת.

## output\_t<int> squad\_duel(int squadId1, int squadId2)

שני צוותי ההאנטרים בעלי המזהים squadId1 ו-squadId2 הסתכסכו והם נלחמים. נסמן את הניסיון של הוצאות squadExp. הילת הקרב אפקטיבית של הוצאה מוגדרת לפ' :

$$\text{squadExp} + \sum_{\text{hunters}} \text{hunter aura}$$

אם לצוות אחד הילת קרבי אפקטיבית גדולה יותר: הוצאה בעל הערך הגדול יותר ניצח. אם שני העריכים שוויים את יכולות הנן של הוצאות. הוצאה בעל יתרון בגין יכולת נ' ניצח (השתמשו באופרטור ההשוואה הלא טרנזיטיבי של NenAbility), ורק אם לשני הוצאות יש יכולת נ' זהה הקרב יסתיים בתיקו. ניצחון מגדייל את squadExp של הוצאה המנצח ב-3, ותיקו מגדייל ערך זה ב-1 לשני הוצאות.

מספר הקרבות שכל אחד מההאנטרים משני הוצאות השתתף בהם גדל ב-1.

## פרמטרים:

מזהה הוצאות הראשון.

squadId1

מזהה הוצאות השני.

squadId2

ערך החזרה: 0 בתיקו; 1 אם הוצאה הראשון ניצח בגין הילת קרבי אפקטיבית; 2 אם הוצאה הראשון ניצח בגין יתרון בגין נ' 3 אם הוצאה השני ניצח בגין הילת קרבי אפקטיבית; 4 אם הוצאה השני ניצח בגין יתרון בגין נ' ובונוס סטטוס:

ALLOCATION\_ERROR

INVALID\_INPUT

FAILURE

SUCCESS

אם אין צוות פעיל עם מזהה 1 או 2 squadId1 או squadId2, או אחד הוצאות ריק.

במקרה של הצלחה.

סיבוכיות זמן:  $O(\log k)$  במקרה הגרוע, כאשר  $k$  מספר הוצאות במערכת.

## output\_t<int> get\_hunter\_fights\_number (int hunterId)

יש להחזיר את מספר הקרבות הכלול שההאנטר בעל המזהה hunterId השתתף בהם, בין אם הוא חי ובין אם נהרג.

## פרמטרים:

מזהה האנטר.

hunterId

ערך החזרה: מספר הקרבות הכלול בהם השתתף ההאנטר, ובונוס סטטוס:

ALLOCATION\_ERROR

INVALID\_INPUT

FAILURE

SUCCESS

במקרה של בעיה בהקצאה/שחרור זיכרון.

.  
אם  $0 \leq \text{hunterId} \leq \text{.hunterId}$

אם מעולם לא היה האנטר עם מזהה hunterId

במקרה של הצלחה.

סיבוכיות זמן:  $O(\log^* n)$  משוערך, ממוצע על הקלט, כאשר  $n$  מספר ההאנטרים שהוכנסו למערכת מעתולם.

**output\_t<int> get\_squad\_experience(int squadId)**

יש להחזיר את הניסיון של צוות האנטרים בעל המזהה `squadId`.

פרמטרים:

<code>squadId</code>	מזהה הצוות.
	<u>ערך החזרה:</u> הניסיון של הצוות <code>squadId</code> , ובנוסף סטטוס:
	ALLOCATION_ERROR     במקרה של בעיה בהקצאה/שחרור זיכרון.
	INVALID_INPUT         אם $0 \leq \text{squadId} < \text{number-of-squads}$ .
	FAILURE                אם אין צוות במערכת עם מזהה <code>squadId</code> .
	SUCCESS                במקרה של הצלחה.
	<u>סיבוכיות זמן:</u> $O(\log k)$ במקרה הגרוע, כאשר $k$ הוא מספר הצוותים במערכת.

**output\_t<int> get\_i\_th\_collective\_aura\_squad(int i)**

נניח שמסדרים את צוותי האנטרים במערכת לפי היליה המשותפת שלהם, כמו למשל:

$$\sum_{\text{hunters}} \text{hunter aura}$$

שובר שוויון עבור צוותים עם אותה היליה משותפת יהיה לפי `squadId` בסדר עולה. יש להחזיר את ה-`i`-po של הצוות במיקום ה-`i` לפי הסדר הזה (אינדקס ראשון = 1).

פרמטרים:

-	<code>i</code> – אינדקס הצוות המבוקש לפי הסדר שהוגדר.
	<u>ערך החזרה:</u> מזהה של הצוות, ובנוסף סטטוס:
	ALLOCATION_ERROR     במקרה של בעיה בהקצאה/שחרור זיכרון.
	INVALID_INPUT         אם אין צוותים במערכת או אם $1 < i \leq \text{number-of-squads}$ .
	FAILURE                במקרה של הצלחה.
	<u>סיבוכיות זמן:</u> $O(\log k)$ במקרה הגרוע, כאשר $k$ הוא מספר הצוותים במערכת.

**output\_t<NenAbility> get\_partial\_nen\_ability(int hunterId)**

רוצים לחשב את יכולת הנק המשותפת של הצוות בו חבר האנטר עם המזהה `hunterId` באופן חלקי בלבד. האנטרים שנלקחים בחשבון בחישוב המזהה `hunterId` והאנטרים שהctrpo לפני (לפי הסדר הכרונולוגי), ולא כל האנטרים במצבם שלו. אם האנטר בעל המזהה `hunterId` הוא האנטר ה-`i` לפי הסדר הכרונולוגי של הצוות שלו, יש לחשב את:

$$\sum_{j=1}^i \text{nenAbility of hunter } j \text{ in the squad of hunterId}$$

שימוש לב: יש להחזיר את אובייקט יכולת הנק עצמו שמתקבל מהסכימה.

פרמטרים:

<code>hunterId</code>	מזהה האנטר.
	<u>ערך החזרה:</u> יכולת הנק המבוקשת, ובנוסף סטטוס:
	ALLOCATION_ERROR     במקרה של בעיה בהקצאה/שחרור זיכרון.
	INVALID_INPUT         אם $0 \leq \text{hunterId} < \text{number-of-squads}$ .
	FAILURE                אם מעולם לא היה האנטר עם מזהה <code>hunterId</code> או שהוא נחרג.
	SUCCESS                במקרה של הצלחה.
	<u>סיבוכיות זמן:</u> $O(n \log^* n)$ משוערך, בממוצע על הקלט.



## StatusType force\_join(int forcingSquadId, int forcedSquadId)

הצאות בעל מזהה forcingSquadId רוצה לכפות על הצאות בעל מזהה forcedSquadId להציגו אליו. צוות יצילח לכפות צוות אחר להציגו אליו אם הסכום של הניסיון, הילה המשותפת יוכלת הנק האפקטיבית שלו (השתמשו בפונקציה getEffectiveNenAbility) גדול ממה משלה אחר (נתיחה לא ריק יכול לעולם אין יכול לכפות, ותמיד יכול להיכפות על ידי צוות שאינו ריק). במקרה, צוות A יוכל לכפות את B להציגו אליו אם:

$A's\ squadExp + A's\ totalAura + A's\ effectiveNenAbility > B's\ squadExp + B's\ totalAura + B's\ effectiveNenAbility$   
הזהה של הצאות הממזוג נשאר זהה למזהה של הצאות הקופה. **הניסיון** של הצאות הממזוג יהיה סכום של הניסיון של שני הצוותים המקוריים. לכל האנטר בצוות הממזוג, **מספר הקרבנות** בהם השתתף נשמר.

פרמטרים:

מזהה צוות כופה.  
מזהה צוות נכפה.

forcingSquadId  
forcedSquadId

ערך החזרה:

במקרה של בעיה בהקצאה/שחרור זיכרון.

ALLOCATION\_ERROR

אם אחד ה-p-o'ים אינם חיובי או שהם שווים.

INVALID\_INPUT

אם אין צוותים עם ה-p-o' הנתונים, או שלא ניתן לבצע את הפעיה כי הצאות הקופה הוא צוות ריק או כי אי השוויון החזק אינו מתקיים.

FAILURE

במקרה של הצלחה.

SUCCESS

**סיבוכיות זמן:**  $O(n^* \log k)$  משוערך, כאשר  $k$  הוא מספר הצוותים במערכת ו- $n$  הוא מספר האנטרים שהיו א' פעם במערכת.



input : output

דוגמה הרצתה:

```

addSquad 1 : SUCCESS
addSquad 2 : SUCCESS
addSquad 3 : SUCCESS
addHunter 10 1 Enhancer 50 0 : SUCCESS
addHunter 11 1 Conjurer 30 2 : SUCCESS
addHunter 12 2 Emitter 40 1 : SUCCESS
addHunter 13 3 Specialist 10 0 : SUCCESS
squadDuel 1 2 : SUCCESS, 1
getHunterFightsNumber 10 : SUCCESS, 1
getPartialNenAbility 11 : SUCCESS, Enhancer:1, Emitter:0, Transmuter:0, Conjurer:1, Manipulator:0, Specialist:0
getSquadExperience 1 : SUCCESS, 3
getIthCollectiveAuraSquad 1 : SUCCESS, 3
forceJoin 1 3 : SUCCESS //squad 3 is no more
getIthCollectiveAuraSquad 1 : SUCCESS, 2
forceJoin 2 1 : FAILURE
removeSquad 3 : FAILURE
addHunter 15 3 Enhancer 20 0 : FAILURE
addSquad -5 : INVALID_INPUT
squadDuel 1 1 : INVALID_INPUT

```

**סיבוכיות מקומית:**

סיבוכיות המקום הדרישה עבור מבנה הנתונים היא  $(k + a) \cdot 0$  במקורה הגראע, כאשר  $a$  הוא מספר האנטרים שוכנסו למערכת מהתחלתה ו- $k$  הוא מספר הקבוצות במערכת מהתחלתה. כלומר לינארית בכמות פעולות ההוספה. סיבוכיות המקום הנדרשת עבור כל פעולה (כלומר, זיכרון "העזר" שככל פועלה משתמש בו) אינה מצינית לכל פעולה לחוד, אך אסור לעבור את סיבוכיות המקום הדרישה שמוגדרת לכל המבנה.

**עלכי החזרה של הפונקציות:**

כל אחת מהfonקציות מחזירה ערך מטיפוס StatusType שייקבע לפי הכלל הבא:

- תחיליה, יוחזר INVALID\_INPUT אם הקלט אינו תקין.
- אם לא הוחזר INVALID\_INPUT:
- בכל שלב בפונקציה, אם קرتה שגיאת הקצאה/שחרור יש להחזיר ERROR\_ALLOCATION\_ERROR. מצב זה אינו צפוי אלא באחד משני מקרים (לרוב): באמצעות השימוש בקלט גדול מאוד ולן המבנה ניצל את כל הזיכרון במערכת, או שיש זליגת זיכרון בקוד.
- אם קرتה שגיאה אחרת, כפי שמצוין בכל פונקציה, יש להחזיר מיד FAILURE\_Memory לשנות את מבנה הנתונים.
- אחרת, יוחזר SUCCESS.
- חלק מהfonקציות צריכות להחזיר בנוסף עוד פרמטר (int או NenAbility), لكن הן מחזירות אובייקט מטיפוס `<T>_output_t`. אובייקט זה מכיל שני שדות: הסטטוס (`status`) ושדה נוספת (`ans`) מסוג `T`.
- במקרה של הצלחה (SUCCESS), השדה הנוסף יכול את ערך החזרה, והסטטוס יכול את SUCCESS. ככל מקרה אחר, הסטטוס יכול את סוג השגיאה והשדה הנוסף לא מעניין.
- שני הטיפוסים (`StatusType`, `output_t<T>`) מומשים כבר בקובץ "h" wet2util.h שנitin لكم כחלק מהתרגום.

**הנחיות וdagשים כלליים:****חלק יבש:**

- החלק הבש הווה חלק מהצין על התרגילים כפי שמצוין בנהלי הקורס.
- לפני שימוש הפעולות בקורס יש לתקן היטב את מבני הנתונים והאלגוריתמים ולודא כי באפשרותכם למש את הפעולות בדרישות הזמן והזיכרונות שלעיל.
- החלק היבש חייב להיות מוקלד.
- הגשת החלק הרטווב מהוות תנאי הכרחי לקבלת ציון על החלק היבש, כמו, הגשה בה יתקבל אך ורק חלק יבש תגרור ציון 0 על התרגיל כולו.
- יש להזכיר מסמך הכוון לתיאור של מבני הנתונים והאלגוריתמים בהם השתמשתם ביצירוף הוכחות סיבוכיות הזמן והמקום שלהם. חלק זה עומד בפני עצמו וצריך להיות מובן לפחות גם לפני העיון בקורס. אין צורך לתאר את הקוד ברמת המסתננים, הפונקציות והמחלקות, אלא בrama העקרונית. חלק יבש איןנו תיעוד קוד.
- 1. ראשית הציגו את מבני הנתונים בהם השתמשתם. הוכחו את דרישות סיבוכיות הזמן של כל פעולה תוך כדי התיחסות לשינויים שהפעולות גורמות לבני הנתונים.
- 2. לאחר מכן הסבירו כיצד מימושם כל אחת מהפעולות במבנה הנתונים. הוכחו את דרישות סיבוכיות הזמן של כל החסמים הנתונים בתרגיל הם לא בהכרח הדוקים ولكن יכול להיות שקיים פתרון בסיבוכיות טובה יותר. מספיק להוכיח את החסמים הדרושים בתרגילים.
- רמת פירוט: יש להסביר את כל הפרטים הבאים טריויאליים וশוחשובים לצורך שימוש הפעולות ועמידה בדרישות הסיבוכיות. יש להסביר שינוי או תוספות בפעולות סטנדרטיות של מבנה נתונים שלמדנו אם נעשו כאלה. אין לדון בפריטים טריויאליים (הפעלו את שיקול דעתכם בקשר לזה, ושאלו את האחראי על התרגיל אם איןכם בטוחים). אין **לצטט קטעים מהקורס כתחליף להסביר.** אין צורך לפרט אלגוריתמים שנלמדו בכתה. כמו כן, אין צורך להוכיח תוצאות ידועות שנלמדו בכתה, אלא מספיק לציין בבירור לאיזו תוצאה אתם מתכוונים. אין (וגם אין צורך) להשתמש בתוצאות של עצי דרגות והלאה.
- **על חלק זה לא לחרוג מ-8 עמודים.**
- והci חשוב **!keep it simple**

**חלק רטווב:**

- **ומומלץ** למשתמש תחיליה את מבני הנתונים באמצעות הכללית ביותר ורק אז **למשתמש** את הפונקציות הנדרשות בתרגילים.
- אלו ממליצים בחום על **שימוש Object Oriented, C++**, מימוש **C++**, מימוש **Object Oriented** יוצר לפונקציות אותן יכולים*ן* למשתמש במבנה נתונים שלם (זכרו שיש תרגיל רטווב נוסף בהמשך הסמסטר).
- פקודות הקימפואל שמורוצת בעמודה gradescope.hינה: g++ -std=c++14 -DNDEBUG -Wall -o main.out \*.cpp
- חתימות הפונקציות שעלייכם למשתמש ומספר הגדרות נמצאים בקובץ h.Huntech26a2.h.
- אין לשנות את הקבצים main26a2.cpp ו-h.h אשר סופקו חלק מהתרגיל, ואין להציג אותם. ישנה בדיקה אוטומטית שאין בקורס שימוש ב-**STL**, ובדיקה זו נופלת אם מגישים גם את cpp.main26a2.
- את שאר הקבצים ניתן לשנות, ותוכלו להוסיף קבצים נוספים כרצונכם, ולהציג אותם.
- העיקר שהקורס שאתם מגישים יתקمال עם הפקודה לעיל, כאשר מושגים לו את שני הקבצים cpp.main26a2 ו-h.wet2util.
- עלייכם למשתמש בעמכם את כל מבני הנתונים (אין **להשתמש** במבנים של **STL** ואין להעתיק מבני נתונים מהאנטרכט). **חלק מתהיליך הבדיקה** אנו נבעז בבדיקה ידנית של הקוד ונודא שאכן מימושם את מבני הנתונים שבמה השמשתם.
- בפרט, אסור להשתמש ב-**iterator**-**shared\_ptr**, **vector**, **pair**, **std::vector**, או כל אלגוריתם של **STL**, רשימה מלאה של הספריות להן אסור להשתמש **include** נמצאת בקובץ dont\_include.txt.
- ניתן להשתמש במצביים חכמים (כמו **shared\_ptr**), בספריית **math** או בספריית **exception**.
- חשוב לוודא שאתם מפעילים זיכרון לצורה נכונה (מומלץ לוודא עם **valgrind**). לא חייבים לעבוד עם מצביעים חכמים, אך אם אתם מפעילים כן לשעות זאת, וודאו שגם שatoms משתמשים בהם נכון. (תזכרו שהם לא פתרון קסם, למשל, כאשר יוצרים מיגל בהצבעות).
- יכולה לגרום ל-**timeout**. עדיף להעביר **shared\_ptr** כשרוצים להעביר **by reference**.
- שגיאות של **ERROR\_ALLOCATION** בד"כ מעידות על זיגזה בזיכרון.
- על הקוד להתקمال ולעבור את כל הבדיקות ש郿ורסות לכם ב-**gradescope**. הטעטים שמורצים באתר מייצגים את הבדיקות אותן נריצ' בנטינת הציון, כאשר פרסמננו 5 מתוך 50.



# מבני נתונים 234218 חורף תשפ"ז

גילון רטוב מס' 2 – מעודכן לתאריך 04.01.2026

עמוד 9 מתוך 9

- אותם טסטים שבסopefile.py גם מפורטים כקבצי קלט ופלט, יחד עם סקריפט בשם tests.py שכתבם בשביל 3.6 python ומעלה, המאפשר לבדוק את הקוד שלכם. מומלץ לבדוק את התרגיל לוקאלית לפני שmagisim.
- צורף קובץ readme.txt למקורה שלא ברור איך להריץ את הטסטים.
- במידה ויש `timeout` על אחד מהטסטים זה יחשב ככשלון בטעט. עלייכם לכתוב קוד יעיל במידת הסביר. אין לדאוג מכך יותר מדי, הרוב המוחלט של הפתרונות שעומד בסיבוכיות עומדת גם בזמן הרצה.
- שימוש:** התוכנית שלכם תידק על קלטיים רבים ושוניים מבחינת הדוגמא הנ"ל. יחד עם זאת, הטסטים האלה מייצגים מבחינת אורך ואופן היוצרת שלהם את השאר.

## אוף ההגשה:

הגשת התרגיל הנה דרך [אתר ה-gradescope של הקורס](#).

### החלק הרטוב:

יש להציג רק את קבצי הקוד שלכם (לרוב קבצי cpp). א.ה. בלבד ואפשר גם להציג אותם כקובץ压缩.

### החלק היבש:

יש להציג קובץ PDF אשר מכיל את הפתרון היבש. החלק היבש חייב להיות מוקלד.

#### שימוש לב Ci אתם מגישים את שני החלקים הנ"ל, במלות השונות.

- לאחר ההגשה, יש באפשרותם לבצע שינויים ולהציג שוב. ההגשה الأخيرة היא זו שת חלה |.
- הערכת הציון שמופיעה ב-gradescope אינה הציון הסופי על המטלה. הציון הסופי יתפרנס רק לאחר הגשות המאוchorות של מתרתי המילואים.
- במידה ואתם חושבים שינוי תקללה מהותית במערכת הבדיקה ב-gradescope נא להעלות זאת בפורום הפיאצה ונטפל בה בהקדם.

## דוחיות ואיתורים בהגשה:

- דוחיות בתרגיל הבית תינטנו ארכויר ורכ לפי תקנון הקורס.
- 5 נקודות יורדו על כל יום איתור בהגשה ללא אישור מראש. באפשרותם להגיש תרגיל באיתור של עד 5 ימים ללא אישור. תרגיל שיוגש באיתור של יותר מ-5 ימים ללא אישור מראש יקבל 0.
- במקרה של איתור בהגשת התרגיל יש עדין להגיש את התרגיל אלקטונית דרך אתר הקורס.
- בקשות להגשה מאוחרת יש להפנות רק למתרגל האחראי: [goldshtein@campus.technion.ac.il](mailto:goldshtein@campus.technion.ac.il).
- לאחר קבלת אישור במיל על הבקשתה, מספר הימים שאושרו לכם נשמר אצלנו. אך, אין צורך לצרף להגשת התרגיל אישורים נוספים או את שער ההגשה באיתור.

**בצלחה!**