

תרגיל מבני נתונים – מסלול DATA

4.9.24 – גרסא 2

תומר שגיא

תוכן עניינים

1.....	תרגיל מבני נתונים – מסלול DATA
1.....	תוכן עניינים
2.....	מטרת הפרויקט
3.....	שלבי ביצוע התוכנה
5.....	בונוס
5.....	הוספת יכולת איזון עץ

מטרת הפרויקט

הפרויקט נועד לבחון את רמת ההבנה של עץ חיפוש בינארי (BST). המטרה שלכם היא לממש עץ חיפוש בינארי מ-0 ע"פ ההנחיות הבאות ולממש פעולות בסיסיות - הוספה, וחיפוש איברים. פעולות נוספות כבנוס לפרויקט, הם איזון העץ ומחיקת איברים. יש שימוש רב ברקורסיות בפרויקט הזה.

הפרויקט שתצטרכו לייצר הוא מערכת למתן מענה להתקפות סייבר, כאשר אתם תקבלו לכם 2 קבצים JSON, אחד מגדיר סוגי התקפות והשני מגדיר סוגי התגוננות למתקפות. כאשר התוכנה תופעל, היא תטען את הקובץ JSON הגנות ותבצע מספר פעולות להכניס את אסטרטגיות ההגנה לעץ חיפוש בינארי (ע"פ ההנחיות למטה) בצורה אופטימאלית. לאחר מכן, המערכת תטען את ה-JSON התקפות ותתחיל להפעיל את ההתקפות אחת אחרי השניה.

כל התקפה כוללת מספר מאפיינים. בכל התקפה, המערכת צריכה לחשב את החומרה (Severity) ע"י נוסחה מסוימת (מופיע בהמשך). לאחר חישוב החומרה (int) המערכת צריכה למצוא את ההגנה המתאימה מעץ ההגנות. כל נקודה בעץ (Node) כוללת טווח חומרות וברגע שחומרת ההתקפה נמצאת בטווח של ה-Node, אסטרטגיית ההגנה תופעל.

אסטרטגיית ההגנה היא בעצם רשימה של מחרוזות. הפעלת אסטרטגיית הגנה כוללת מעבר על רשימת ההגנות והדפסתם אחת אחרי השניה תוך כדי המתנה בין הדפסה להדפסה (כדי לתת תחושה של ביצוע פעולה שלוקחת זמן).

לאחר בניית העץ, אתם צריכים לבצע רצף של פעולות, **בדיוק כמו שכתוב בדף הבא**, אחת אחרי השניה.

מה אתם מקבלים:

- קובץ **threats.json** – קובץ התקפות
- קובץ **defenceStrategiesBalanced.json** – קובץ אסטרטגיות הגנה ע"י Severity
- קובץ **defenceStrategies.json** – קובץ אסטרטגיות הגנה לשאלת בנוס (עץ לא מאוזן)

בשלב הראשון של הפרויקט עליכם לבנות מבנה נתונים של עץ חיפוש בינארי (BST) כאשר כל Node בעץ יכול לומר את הנתונים הבאים:

1. טווח חומרה (MinSeverity ו-MaxSeverity).

2. `<List<string>` של הגנות.

שלבי ביצוע התוכנה

התוכנה צריכה לבצע את רצף הפעולות הבאות (בדף הבא) בין כל פעולה יש צורך לשים השהייה של 4 שניות ולהדפיס את השלב שמתבצע כרגע לקונסול.

1. יצירת העץ הבינארי ע"י ייבוא קובץ JSON עם טווחי חומרה והכנסה לעץ

1. יבא את קובץ JSON המכיל טווחי חומרה בסדר עולה –

defenceStrategiesBalanced.json

2. מימוש מתודת Insert במחלקה DefenceStrategiesBST כדי להוסיף צמתים לעץ.

3. הכנס את כל הצמתים לעץ באמצעות המתודה Insert.

2. הדפסת העץ בצורה של עץ – PreOrder Traversal

1. מימוש מתודה להדפסת העץ – כאן צריך להשתמש במעבר על העץ בצורה של

Pre-Order Traversal

2. הדפסת העץ (דוגמא למטה)

```
Tree structure with left/right child distinctions:
Root: [15-20] Defenses: Rate-limit IPs, Activate firewall
├─Left Child: [5-10] Defenses: Monitor traffic, Log incident
└─Right Child: [25-30] Defenses: Isolate system, Deploy malware scan
    └─Right Child: [35-40] Defenses: Activate IDS, Notify security team
```

3. ייבוא איומים מקובץ JSON

1. השתמש בקובץ threats.json – JSON

2. יבא איומים אלה לתוכנית שלך.

4. מימוש חישוב חומרה של התקפה

1. ממש את נוסחת חישוב חומרת כל איום בהתבסס על מאפייניו.

הנוסחה לחישוב Severity היא להלן:

$$\text{int severity} = (\text{Volume} * \text{Sophistication}) + \text{TargetValue};$$

כאשר ההמרה של TargetValue למספר יעשה ע"י המיפוי הבא:

"Web Server" => 10,

"Database" => 15,

"User Credentials" => 20,

כל אחד אחר <= 5

5. הטמעת מתודת חיפוש להגנות והפעלת ההגנות אחת אחרי השניה

1. יישם שיטת חיפוש באמצעות Pre-Order Traversal כדי למצוא את הצומת בעץ

המתאים לחומרה המחושבת.

2. הפעל את רשימת ההגנות עבור כל איום בהתבסס על תוצאות החיפוש – הכוונה רק להדפיס אותם אחת אחרי השניה תוך שימוש **בהשהייה של 2 שניות ביניהם**
3. במידה והחומרה נמוכה מהמינימום, צריכה להופיע הודעה "Attack severity is below the threshold. Attack is ignored"
4. במידה והחומרה לא נמצאת בעץ, תודפס ההודעה "No suitable defence was found. Brace for impact"

בונוס

הוספת יכולת איזון עץ

השלבים הבאים יחליפו את שלבים 1 ו-2 מתאור הפרויקט המקורי, יתר השלבים ישארו כמו שהם

1. **יצירת העץ הבינארי ע"י ייבוא קובץ JSON עם טווחי חומרה והכנסה לעץ**
 1. יבא את קובץ JSON המכיל טווחי חומרה בסדר עולה –
defenceStrategies.json
2. **הדפסת העץ בצורה של עץ – PreOrder Traversal**
 1. כאן נראה שהעץ לא מאוזן
3. **מימוש מתודת BalanceTree והדפסת העץ – 5 PreOrder Traversal נקודות]**
 1. ליישם מתודה לאזן את העץ שתהיה מאוזנת הכי שאפשר
 2. ודאו שהעץ נשאר מאוזן לאחר האיזון ע"י הדפסת העץ
4. **הדפיסו את העץ כרשימה – InOrder Traversal לראות שהסדר נשמר**
 1. הדפסת העץ כדי לראות שהוא מאוזן – הדפסה כרשימה
 2. יישם פעולת שירות InOrderTraversal כדי להדפיס את טווחי החומרה ואת ההגנות התואמות להם לפי הסדר – כדי לראות שהסדר עדין נשמר נכון
5. **ייצא את העץ המאוזן לקובץ JSON**
 1. הפוך את מבנה העץ המאוזן בסידרה בחזרה לקובץ JSON.
 2. יש צורך כאן להשתמש בלוגיקה של מעבר Pre-Order כדי לשמור את ה-JSON לפי הצורה המאוזנת