דגשים חשובים לבחינה במבני נתונים

---

1. רשימה מקושרת (Node)

- **שמירה על הראש:** אין "לטייל" עם המצביע המקורי של ראש הרשימה. תמיד נגדיר מצביע עזר: Node<T> pos = first
- **סריקת רשימה:** הבדיקה היא כל עוד pos =! null. אם ניגשים בכל איטרציה לחוליה הבאה אחרי pos, הבדיקה תהיה כל עוד pos.GetNext() =! null.
- **שינוי רשימה:** בפעולה שמוסיפה או מוחקת איברים, יש להחזיר את ראש הרשימה החדש למקרה שהשינוי התבצע בתחילת הרשימה.
- **ערך "זבל":** בבניית רשימה חדשה, ניתן להשתמש בחוליית דמה כדי לפשט את הלוגיקה, ובסיום להחזיר את list.GetNext().

2. תור - FIFO ראשון נכנס, ראשון יוצא (Queue)

- **שמירה על הערכים:** כדי לגשת לאיבר שאינו בראש התור, חייבים להוציא איברים. יש להשתמש בתור עזר ולהחזיר את האיברים בסיום כדי שהתור המקורי לא יימחק.
- **סדר האיברים:** כדי לשמור על סדר התור המקורי, יש לרוקן את כל האיברים שנותרו בתור המקורי לתור העזר, ורק אז להחזיר את כולם חזרה.
- **לא לגשת ליותר מאיבר אחד בכל איטרציה ללא בדיקה:** לפני ביצוע Remove() או Head(), חובה לוודא שהתור אינו ריק בעזרת IsEmpty().

3. מחסנית - LIFO אחרון נכנס, ראשון יוצא (Stack)

- **סדר הפוך:** המחסנית מתנהגת כמו "ערימת צלחות". האיבר האחרון שנכנס הוא הראשון שיוצא.
- **שמירה על המחסנית:** בדומה לתור, אם נרצה לעבור על איברי המחסנית מבלי להרוס אותה, נשתמש במחסנית עזר.
  - **שימי לב:** העברה ממחסנית אחת לאחרת הופכת את סדר האיברים. כדי להחזיר את המחסנית למצבה המקורי, יש להעביר את האיברים למחסנית העזר, ובסיום להעבירם חזרה למקורית.
- **פעולות יסוד:**
  - Pop(): מוציאה ומחזירה את הערך העליון.
  - Top(): רק מציצה בערך העליון בלי להוציא אותו.
- **לא לגשת ליותר מאיבר אחד בכל איטרציה ללא בדיקה:** לעולם אין לבצע Pop() או Top() על מחסנית ריקה. תמיד לבדוק IsEmpty().

4. עץ בינארי (BinNode)

- **רקורסיה:** רוב הפעולות על עצים יתבצעו בעזרת רקורסיה.
- **מקרה קצה (תנאי עצירה):** תמיד לבדוק תחילה אם העץ ריק: if (t == null).
- **גישה לבנים:** לפני פנייה ל-t.GetLeft() או t.GetRight(), יש לוודא שהבן קיים בעזרת HasLeft() או HasRight(), או לוודא שהצומת t עצמו אינו null.
- **סריקות:**
  - **In-order (תוכי):** שמאל -> שורש -> ימין.
  - **Pre-order (תחילי):** שורש -> שמאל -> ימין.
  - **Post-order (סופי):** שמאל -> ימין -> שורש.
- **עץ עלה:** צומת הוא עלה אם:() t.HasRight! && ()t.HasLeft!.

5. דגשים כלליים לכתיבת קוד

- **טיפוסים:** השוואה (==) או השמה מתבצעת רק בין משתנים מאותו טיפוס (למשל: השוואת T ל-T, או T>Node< ל-T>Node<).
- השוואה בין עצמים היא השוואה בין ההפניות, ולא בין ערכי התכונות שלהם. לדו', if(pos1==pos2) זו בדיקה האם שני המצביעים מצביעים לאותה צומת, ולא אם ערכיהם זהים.