



# הגרסא האדפטיבית-דינאמית לאלגוריתם הכיווץ של הופמן ישראל אור

מסמך זה הורד מהאתר <http://www.underwar.co.il>

מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר. המחבר יצא מנקודת הנחה שאלגוריתם הכיווץ הסטטי של הופמן אינו זר לקורא.

כל הזכויות שמורות ל-ישראל אור

## הקדמה:

כמעט מייד לאחר שמימשתי לראשונה את אלגוריתם הופמן, וראיתי את הפשטות היחסית בה ניתן לממש את האלגוריתם בהשוואה לתועלת האדירה כתוצאה מהיכולת לשמור יותר בפחות מקום ואת הפוטנציאל העצום בשילוב כלי זה עם תוכניות אחרות שכתבתי, התחלתי לחשוב על אפשרות לשלב את האלגוריתם בתקשורת עם שני זרמי נתונים נפרדים.

היתרון הוא הקטנת נפח המידע בזמן אמת, כאשר זרם אחד שולח מידע דחוס והשני פורס אותו. הפתרון המוצע במאמר מסייע להתמודד עם העובדות הבאות:

1. ב- 99% מהמקרים של זרימת מידע בזמן אמת הנמען לא יכול לצפות מראש את המידע שיעביר המוען, הדבר לא מאפשר יצירת טבלת תדירויות קבועה בהסתמך על מידע שאין ברשותו של הנמען.
2. ללא טבלת תדירויות לא ניתן לפרוס את זרם המידע המתקבל.
3. בניית טבלת תדירויות והעברתה עם זרם הנתונים תעלה בזמן יקר. דבר שאינו מתקבל על הדעת, שלא לדבר על העובדה שהתוצאה תהיה ניפוח כמות המידע.

להלן מוצע הפתרון:

## הגרסא הדינאמית של אלגוריתם הכיווץ של הופמן

ישנן מספר שיטות ליישם את האלגוריתם הדינאמי, שתי השיטות עיקריות הן: **Vitter algorithm** - **FGK** (Faller-Gallager-Knuth) ו- **Vitter**. במדריך נממש את שיטת **Vitter**.

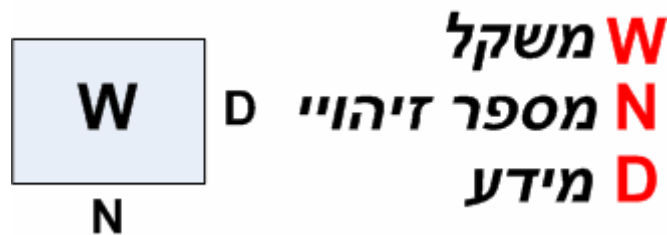
גרסא זו דומה למדי לגרסא הסטטית עם מספר שינויים:

1. אין טבלת תדירויות.
2. עץ התדירויות נבנה ומתעדכן בצורה דינאמית בהתאם לזרם הנתונים.

הרעיון הוא לשמור שעץ התדירויות יהיה זהה, הן בצד המקבל והן בצד השולח וישמור תמיד על חוקים מסוימים, לדוג' הצומת עם ציון תדירות (משקל) תמיד יהיה אב לצומת עם משקל נמוך יותר (כאשר בעץ לפחות צומת אחד) - בהמשך אפרט את הכללים.

### המושגים בהם משתמש המדריך

- **משקל :** מייצג את תדירות הופעת תו בזרם המידע, כל צומת בעץ הינו בעל משקל. צומת שאינו עלה מורכב מסכום משקלי בניו.
- **בלוק:** קבוצת הצמתים בעץ בעלי אותו המשקל.
- **מ"ז צומת:** כל צומת מצויד במספר מזהה ייחודי שישמש אותנו לזיהוי מהיר, לארגון\איזון העץ ולצרכי התייחסות לצומת.
- **מידע:** כל העלים בעץ יחזיקו יחידת מידע (מלבד צומת NYT) כדוגמת תו בצורה לא דחוסה, כאשר בצורה דחוסה התו ייוצג ע"י רצף הפניות משורש העץ ועד לצומת המחזיק את התו כאשר כל פניה (ימינה או שמאלה) מיוצגת ע"י 1\0.
- **NYT:** Not-Yet-Transmitted צומת עלה הקיים תמיד, בכל עץ (גם בעץ התחלתי ומכאן שלא יתכן מצב של עץ ריק), בעל משקל 0 ותפקידו לסמן את המקום הבא אליו יתווסף צומת חדש (צומת יתווסף רק במידה ותו חדש יעבור בזרם הנתונים).



### מסקנות עד כה

1. לצומת בן תמיד יהיה משקל נמוך ממשקל אביו.
2. שורש העץ יהיה בעל המשקל הגבוה ביותר.
3. צומת עם משקל גבוה יותר יצוין ע"י פחות תנועות במורד העץ. מכאן – המידע המיוצג ע"י הצומת יהיה דחוס יותר אחרי כיווץ.
4. אותן פעולות מתבצעות על העץ הן בצד השולח והן בצד המקבל (המפתח לשמירת עץ זהה).

## הבהרות

1. כאשר נשלח מידע קיים בעץ הוא מועבר בצורה דחוסה, עץ הופמן מתעדכן עם המשקלים החדשים ושומר על איזון.
  2. כאשר נשלח מידע חדש, הוא מועבר כמות שהוא (ללא דחיסה), ונוסף לעץ צומת חדש עם המידע. במקום צומת ה-NYT, נוצר צומת NYT חדש והעץ מעדכן את המשקלים ומתאזן.
- \* מובן שהקושי הוא בשמירת איזון העץ ככל הניתן, ועיקר האלגוריתם הוא בפתירת הבעיה.

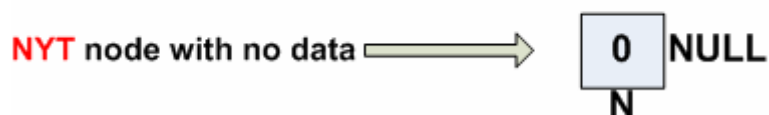
## עץ ראשוני

כאמור, מבנה העץ הן בצד השולח והן בצד המקבל זהה לחלוטין, הוא הדין גם לגבי העץ הראשוני.

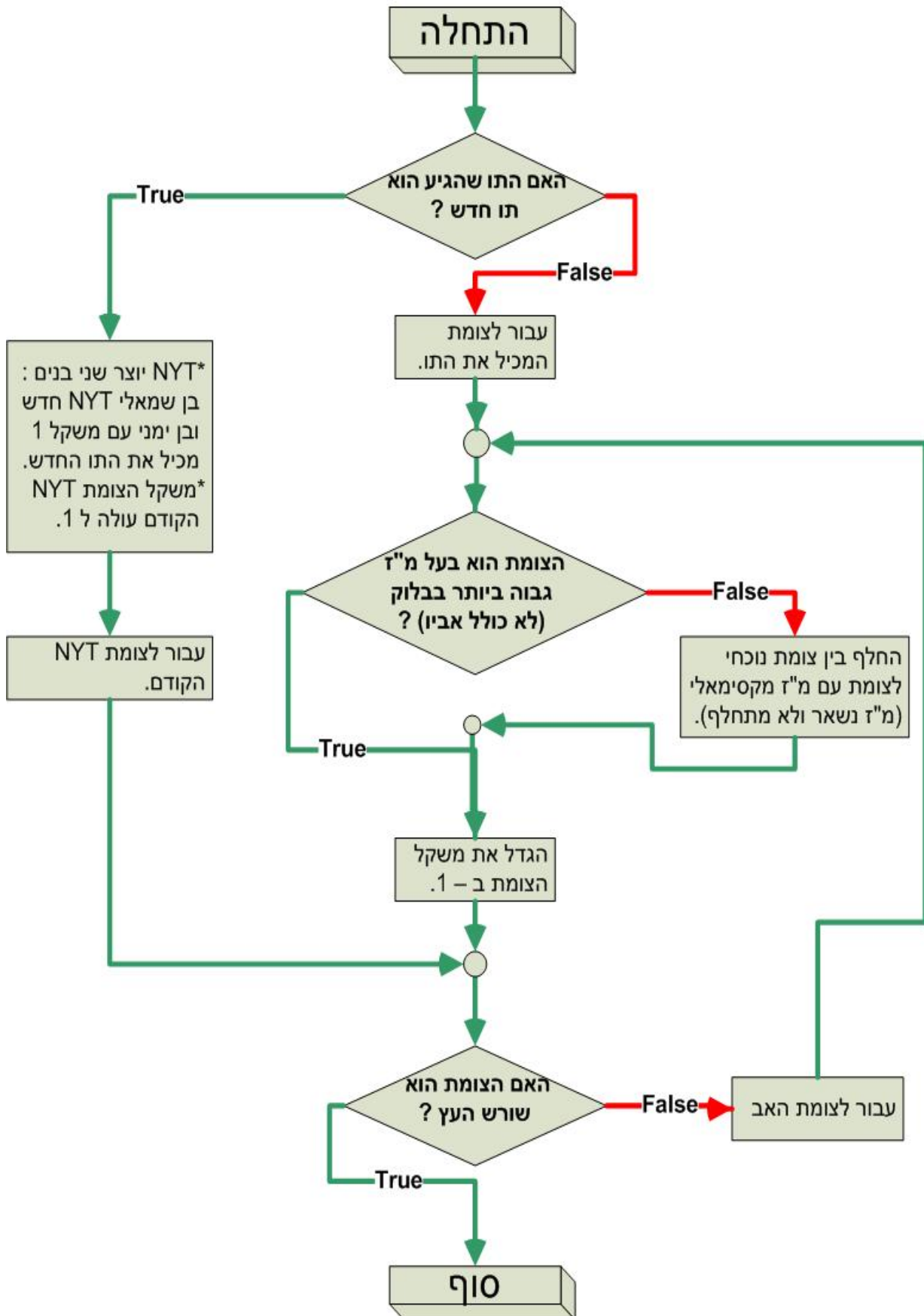
העץ הראשוני מורכב מצומת NYT אחד עם משקל 0 ו מספר זיהוי N כאשר:  
עם כל הוספת צומת לעץ, מספר הזיהוי של הצומת החדש יהיה N-1 (אחת פחות מספר הזיהוי של צומת ה-NYT הנוכחי) ומספר הזיהוי של צומת ה-NYT החדש יהיה N-2.

מאחר והוספת מידע חדש תעלה בהוספת 2 צמתי עלים חדשים (אחד למידע ואחד NYT חדש), מספר העלים מכילי התווים המקסימאלי הוא N/2.

## **סכמת עץ חדש:**



האלגוריתם כתרשים זרימה



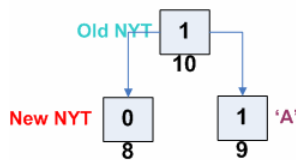
**הדגמת האלגוריתם (מומלץ לעקוב אחר התרשים בדף הקודם)**



נתחיל עם עץ חדש:

1. נאמר שנשלח אלינו מידע חדש (לא מכווץ) בצורת התו 'A':
  - ניצור שני בנים ל-NYT הקיים כשאחד מהם יהיה NYT חדש.
  - נמקם את צומת ה-NYT החדש כבן שמאלי ואת הצומת שיכיל את התו כבן ימני.
  - מ"ז בן ימני יהיה מ"ז צומת NYT קודם פחות 1 (9), מ"ז בן שמאלי יהיה מ"ז צומת NYT קודם פחות 2 (8).
  - משקל הצומת שיכיל את התו כבן ימני יהיה 1 (התו הופיע פעם אחת עד כה).
  - נעדכן את משקל צומת ה-NYT הקודם, כלומר נגדיל אותו ב-1 (מ-0 ל-1).
  - נעבור לצומת NYT הקודם, האם הוא שורש? כן!, לכן הסתיים הליך ההוספה.

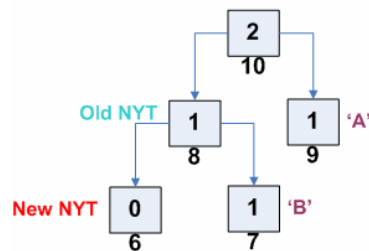
**העץ לאחר הוספת התו A**



2. נוסף את התו 'B' עם אותו סדר פעולות:

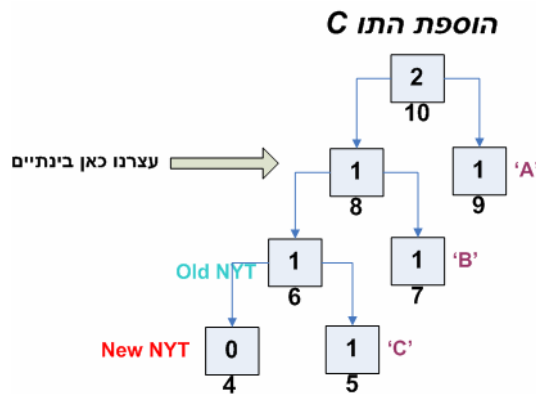
- ניצור שני בנים ל-NYT הקיים כשאחד מהם יהיה NYT חדש.
- נמקם את צומת ה-NYT החדש כבן שמאלי ואת הצומת שיכיל את התו כבן ימני.
- מ"ז בן ימני יהיה מ"ז צומת NYT קודם פחות 1 (7), מ"ז בן שמאלי יהיה מ"ז צומת NYT קודם פחות 2 (6).
- משקל הצומת שיכיל את התו כבן ימני יהיה 1 (התו הופיע פעם אחת עד כה).
- נעדכן את משקל צומת ה-NYT הקודם, כלומר נגדיל אותו ב-1 (מ-0 ל-1).
- נעבור לצומת NYT הקודם, האם הוא שורש? לא! לכן נמשיך:
- נעבור לצומת האב (מ"ז 10). האם מ"ז שלו הוא הגדול ביותר בבלוק (לא כולל אביו, אם קיים)? **כן!** – נגדיל את משקל הצומת ב-1 (כעת משקל צומת השורש גדל מ-1 ל-2), האם הוא שורש? **כן!**, לכן הסתיים הליך ההוספה.

**העץ לאחר הוספת התו B**



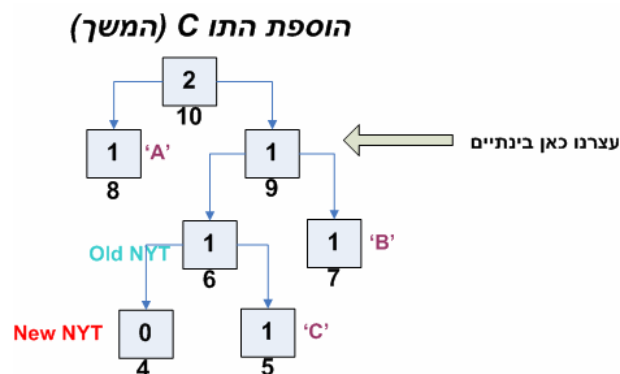
3. נוסף את התו 'C' באותו סדר פעולות:

- ניצור שני בנים ל-NYT הקיים כשאחד מהם יהיה NYT חדש.
- נמקם את צומת ה-NYT החדש כבן שמאלי ואת הצומת שיכיל את התו כבן ימני.
- מ"ז בן ימני יהיה מ"ז צומת NYT קודם פחות 1 (5), מ"ז בן שמאלי יהיה מ"ז צומת NYT קודם פחות 2 (4).
- משקל הצומת שיכיל את התו כבן ימני יהיה 1 (התו הופיע פעם אחת עד כה).
- נעדכן את משקל צומת ה-NYT הקודם, כלומר נגדיל אותו ב-1 (מ-0 ל-1).
- נעבור לצומת NYT הקודם (מ"ז 6), האם הוא שורש? לא! לכן נמשיך:
- נעבור לצומת האב (מ"ז 8).



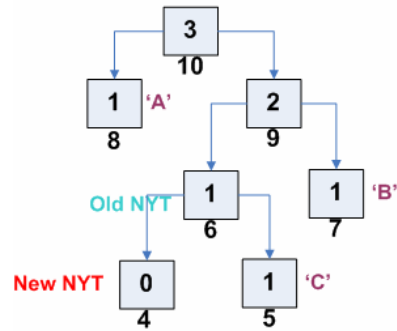
**כללי החלפות בין צמתים:**  
 - החלפה תתאפשר רק בין צמתים השייכים לאותו הבלוק.  
 - החלפה לא תבצע בין צומת אב לצומת בן.  
 - כאשר צמתים מתחלפים, כל תת העצים הקשורים להם נעים איתם.  
 - כאשר צמתים מתחלפים מ"ז לא מתחלף עם ההזזה.

- האם מ"ז שלו הוא הגדול ביותר בבלוק (לא כולל אביו אם קיים)? לא! הצומת עם מ"ז 9 אף הוא באותו בלוק ואינו אביו של צומת נוכחי (מ"ז 8), לכן נחליף בין שני תתי העצים:



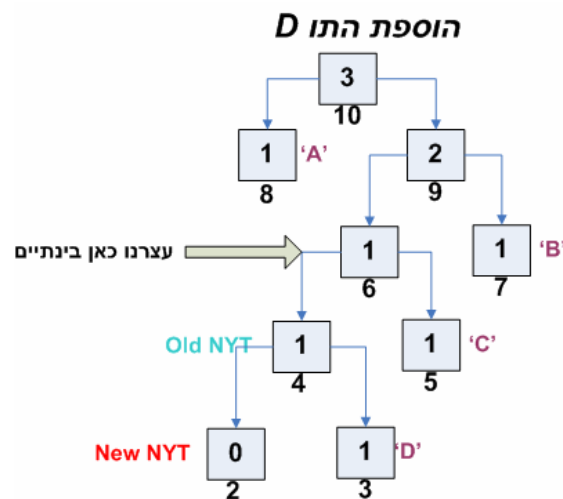
- נגדיל את משקל הצומת הנוכחי (מ"ז 9) ב-1 (כעת משקלו 2). האם הוא שורש? **לא!** לכן נמשיך לצומת האב:
- האם מ"ז שלו הוא הגדול ביותר בבלוק (לא כולל אביו אם קיים)? **כן!** – הגדיל את משקל הצומת ב-1 (כעת משקל צומת השורש גדל מ-2 ל-3), האם הוא שורש? **כן!** לכן הסתיים הליך ההוספה.

**הוספת התו C (סיום)**

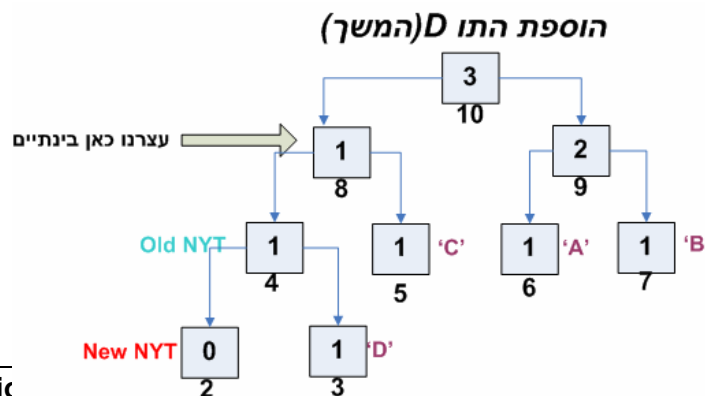


4. נוסיף את התו 'D' באותו סדר פעולות:

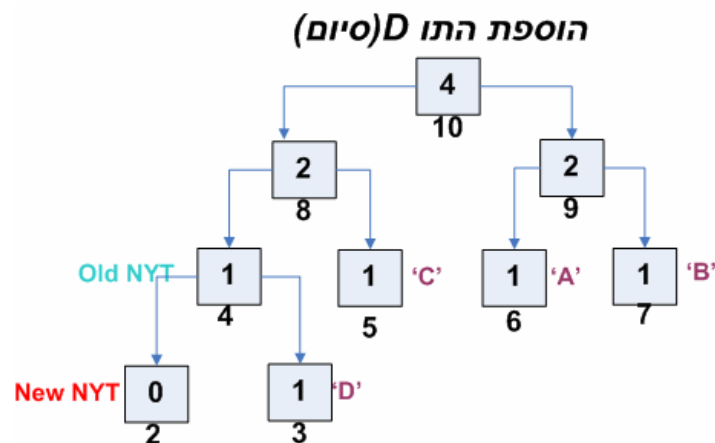
- ניצור שני בנים ל-NYT הקיים כשאחד מהם יהיה NYT חדש.
- נמקם את צומת ה-NYT החדש כבן שמאלי ואת הצומת שיכיל את התו כבן ימני.
- מ"ז בן ימני יהיה מ"ז צומת NYT קודם פחות 1 (3), מ"ז בן שמאלי יהיה מ"ז צומת NYT קודם פחות 2 (2).
- משקל הצומת שיכיל את התו כבן ימני יהיה 1 (התו הופיע פעם אחת עד כה).
- נעדכן את משקל צומת ה-NYT הקודם, כלומר נגדיל אותו ב-1 (מ-0 ל-1).
- נעבור לצומת NYT הקודם (מ"ז 4). האם הוא שורש? לא! לכן נמשיך:
- נעבור לצומת האב (מ"ז 6).



- האם מ"ז שלו הוא הגדול ביותר בבלוק (לא כולל אביו, אם קיים)? לא!
- הצומת עם מ"ז 8 אף הוא באותו בלוק ואינו אביו של צומת נוכחי (מ"ז 6), לכן נחליף בין שני תתי העצים:



- נגדיל את משקל הצומת הנוכחי (מ"ז 8) ב-1 (כעת משקלו 2). האם הוא שורש? **לא!** לכן נמשיך לצומת האב:
- האם מ"ז של הצומת (מ"ז 10) הוא הגדול ביותר בבלוק (לא כולל אביו, אם קיים)? **כן!** – נגדיל את משקל הצומת ב-1 (כעת משקל צומת השורש גדל מ 3 ל-4), האם הוא שורש? **כן!** לכן הסתיים הליך ההוספה.

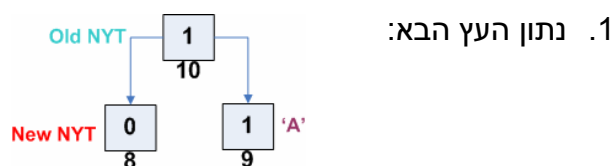


### העברת מידע קיים

כל הדוגמאות עד כה היו של הוספת תו חדש, **כעת נדגים מקרה של העברת מידע קיים** (במקרה שלנו תו) בצורה מכווצת.

כאמור, כאשר עובר מידע קיים (יעבור תמיד בצורה מכווצת), יש לעדכן את משקלו בצומת המתאים. **לפני** עדכון משקלו (הגדלתו ב-1), נוודא שהצומת המדובר בעל מ"ז מקסימאלי בקבוצת הבלוק שלו, אחרת נבצע החלפה בדיוק כמו שבצענו עד עכשיו...

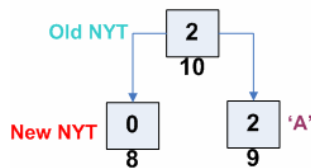
*\*מומלץ שוב לעקוב אחר תרשים זרימת האלגוריתם*



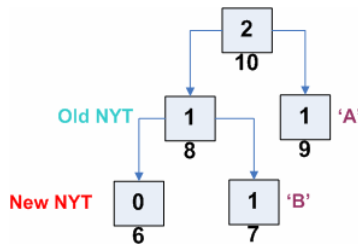
-נתון שמידע מכווץ עם התו 'A' עבר **שוב** בזרם הנתונים.

- נחפש את הצומת שמכיל את אותו המידע (מ"ז 9).
- האם מ"ז של הצומת (מ"ז 9) הוא הגדול ביותר בבלוק (לא כולל אביו אם קיים)? **כן!** – נגדיל את משקל הצומת ב-1 (כעת משקל הצומת גדל מ-1 ל-2). האם הוא שורש? **לא!** לכן נמשיך לצומת אב:
- האם מ"ז של הצומת (מ"ז 10) הוא הגדול ביותר בבלוק (לא כולל אביו, אם קיים)? **כן!** – נגדיל את משקל הצומת ב-1 (כעת משקל הצומת השורש גדל מ-1 ל-2). האם הוא שורש? **כן!** לכן הסתיים תהליך עדכון.

**העץ לאחר עדכון התו A**



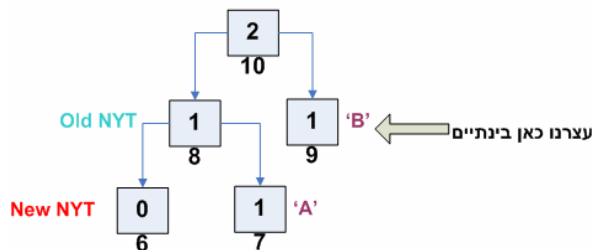
2. נתון העץ הבא:



-נתון שמידע מכווץ עם התו 'B' עבר **שוב** בזרם הנתונים.

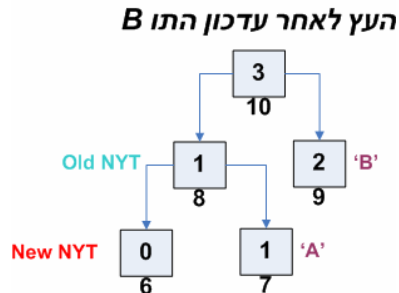
- נחפש את הצומת שמכיל את אותו המידע (מ"ז 7).
- האם מ"ז של הצומת (מ"ז 7) הוא הגדול ביותר בבלוק (לא כולל אביו, אם קיים)? **לא!** הצומת עם מ"ז 9 אף הוא באותו בלוק ואינו אביו של צומת נוכחי (מ"ז 7), לכן נחליף בין שני תתי העצים:

**עדכון העץ עם התו B (המשך)**



- נגדיל את משקל הצומת הנוכחי (מ"ז 9) ב-1 (כעת משקלו 2). האם הוא שורש? **לא!** לכן נמשיך לצומת האב:

- האם מ"ז של הצומת (מ"ז 10) הוא הגדול ביותר בבלוק (לא כולל אביוף אם קיים)? **כן!** – נגדיל את משקל הצומת ב-1 (כעת משקל צומת השורש גדל מ 2 ל-3), האם הוא שורש? **כן!** לכן הסתיים הליך ההוספה.



### זמנים ומספרים

\*עלות פעולות בעץ היא  $\sim 2t+S$

- כאשר  $t$  היא עלות בעץ הופמן סטטי ו- $S$  הוא אורך התווים בהודעה המקורית.

\*גודל תו דחוס במקרה הממוצע יהיה  $[Log_2 n]$  כאשר  $n$  הוא אורך ההודעה בתווים.