אתה יכול להגיד את זה שוב! - דחיסת טקסט

תקציר

כיוון שלמחשבים יש כמות מוגבלת של שטח אחסון מידע, הם צריכים לייצג מידע בצורה יעילה ככל האפשר, זה נקרא דחיסה. על ידי קידוד נתונים לפני שהם מאוחסן, ופענוחם כאשר הם מוחזרים, המחשב יכול לאחסן יותר נתונים, או לשלוח אותם מהר יותר דרך האינטרנט.

מונחים טכניים

דחיסת טקסט, קידוד למפל-זיו, קידוד האפמן.

חומרים

<u>כל ילד צריך :</u>

עותק של דפי הפעילות.

מבוא

מחשבים צריכים לאחסן ולהעביר כמות גדולה של נתונים, כדי שהם לא יצטרכו להשתמש בהרבה מדי מקום אחסון, או לקחת זמן רב מדי כדי לשלוח מידע באמצעות חיבור מודם, הם דוחסים את הטקסט בצורה דומה לזו.

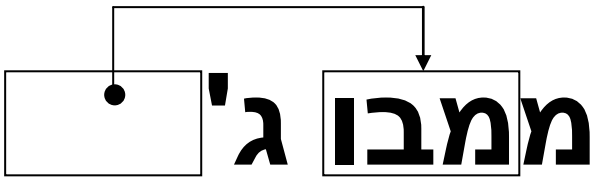
הדגמה ודיון

הראו לכיתה את השיר בעמוד 2.

דיון - הסתכלו על התבניות של האותיות בשיר, האם אתם מזהים קבוצה של 2 אותיות ומעלה שחוזרות על עצמן, או אפילו מילים או משפטים שלמים?

החליפו אותם בתיבות כפי שמוצג בתרשים הבא:

ממבו ג'מבו



לאחר ההדגמה יש לחלק את **דף פעילות 1**, ניתן להכין טקסטים ושירים מראש ולחלק בין הילדים עבור החלק השני של הפעילות בו הם צריכים להכין פאזל לבד או לתת להם להשתמש בדמיון ☺

ממבו ג'מבו

ממבו ג'מבו

כולם רוקדים עכשיו

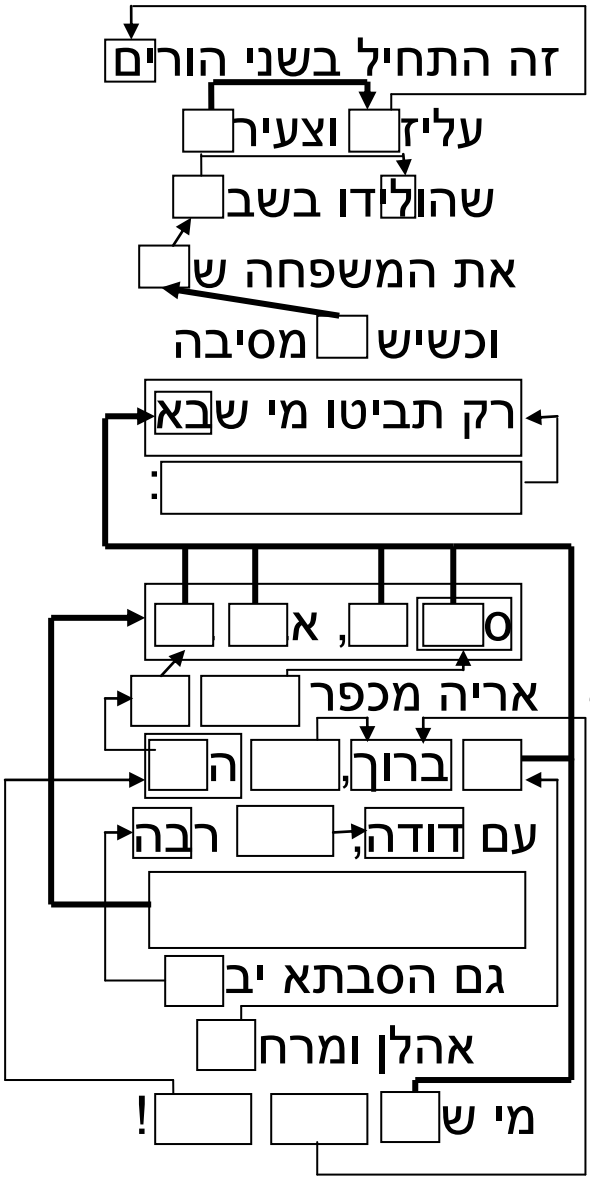
ממבו ג'מבו

ממבו ג'מבו

גם אתה כוכב!

דף פעילות 1: אתה יכול להגיד את זה שוב!

אותיות ומילים רבות חסרות בשיר הזה. האם ניתן למלא את המילים והאותיות החסרות בצורה נכונה?
תמצאו את התשובות בתיבות שהחצים מצביעים עליהן☺



כעת בחרו שיר כרצונכם ותעצבו חידון משלכם!

שימו לב שהחצים תמיד יצביעו לחלק מוקדם יותר בטקסט, השיר צריך להיות מפוענח מימין לשמאל ומלמעלה למטה באותה צורה שאנחנו קוראים.

אתגר: נסו לשמור כמה שפחות מהמילים המקוריות!

הצעות: נסו להימנע מצפיפות יתר של חצים. השאירו הרבה מקום סביב אותיות ומילים תוך כדי כתיבה כך שיישאר מקום לתיבות בתוך תיבות ולחצים המצביעים אליהן.
קל יותר לעצב את החידון אם כותבים את השיר קודם ואז מחליטים איפה התיבות צריכות להיות.

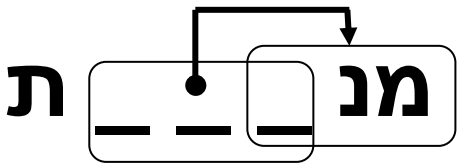
פתרון לדף פעילות 1:

זה התחיל בשני הורים
עליזים וצעירים
שהולידו בשבילי
את המשפחה שלי
וכשיש לי מסיבה
רק תביטו מי שבא
רק תביטו מי שבא:

סבא בא, אבא בא
אריה מכפר סבא בא
בא ברוך, ברוך הבא
עם דודה, דודה רבה
סבא בא, אבא בא
גם הסבתא יבבה
אהלן ומרחבא
מי שבא ברוך הבא!

פעילות 2 – קידוד למתקדמים:

ציירו על הלוח/ הדפיסו את החידה כפי שמוצגת למטה ושאלו את הילדים: איך נפתור את החידה הבאה?



הסבר:
לפעמים טקסט חסר מצביע לעצמו. במקרה הזה ניתן לפענח את המילה בצורה נכונה אם נעתיק את האותיות מימין לשמאל, ואז כל אות זמינה להיות מועתקת לפני שהיא נחוצה. האפשרות הזאת שימושית במחשבים אם יש רצף ארוך של אות או תבנית מסוימת.

במחשב התיבות והחיצים מיוצגים על ידי מספרים, לדוגמה : **מנמנמת**

ניתן לכתוב כ: **מנ(2,3)ת**. **"2"** מייצג – קפוץ אחורה 2 אותיות לנקודת ההתחלה להעתקה:

מנ_ _ _ת

ו **"3"** אומר – העתק 3 אותיות ברצף:

מנמ_ _ת

מנמנ_ת

מנמנת

כיוון ששני מספרים משמשים אותנו לקידוד המילים, לרוב נרצה לדחוס רצפים של שתי אותיות ומעלה, אחרת אין לנו חיסכון בשטח. למעשה, הגודל של הקובץ יכול לעלות אם שני מספרים נמצאים בשימוש עבור קידוד של אות אחת.

כעת עודדו אותם להמציא ולקודד מילים משלהם ולהעביר לחברים לפענוח ☺

מילים לדוגמא: פרפר, כלכלה, שרשרת, גולגולת, מצמצמים, אותות, אפיפית, לקלקל.

כמה מילים באמת צריך פה?

דמיינו שאתם מחשב שמנסה לדחוס כמה שיותר מידע לזיכרון. מחקו את כל הרצפים של שתי אותיות ומעלה שכבר הופיעו בעבר. רצפים אלו מיותרים כעת וניתן להחליף אותם על ידי מצביע. המטרה שלכם היא למחוק כמה שיותר אותיות.

שוּלָה שׁוֹלָה שַׁבְּלוּלִים בַּשְּׁלוּלִית.

כֹּל שַׁבְּלוּל אֲשֶׁר שׁוּלָה שׁוֹלָה הוּא שֶׁלָּהּ.

שֶׁלָּהּ כֹּל שַׁבְּלוּל אֲשֶׁר שׁוּלָה שׁוֹלָה

בַּשְּׁלוּלִית שֶׁבָּהּ שׁוּלָה שׁוֹלָה שַׁבְּלוּלִים.

התמונה הגדולה

קיבולת האחסון של מחשבים גדלה בשיעור לא יאומן ב-25 השנים האחרונות, כמות האחסון שסופקה במחשב טיפוסי גדלה פי מליון! אבל עדיין אנחנו מוצאים עוד דברים להכניס לזיכרון של המחשב. מחשב יכול לאחסן ספרים שלמים ואפילו ספריות וכעת גם מוזיקה וסרטים במידה ויש מספיק מקום. קבצים גדולים יוצרים בעיה גם באינטרנט כיוון שלוקח הרבה זמן להוריד אותם, מה גם שאנחנו כל הזמן מנסים להקטין את הגודל של המחשבים – היום אפילו טלפון נייד או שעון יד מאחסנים מידע רב.

יש פתרון לבעיה זו, עם זאת, במקום לקנות עוד שטח אחסון או מודם מהיר יותר, אנחנו יכולים לדחוס את המידע כך שהוא יתפוס פחות מקום. תהליך זה של דחיסה ושחזור הנתונים נעשה בדרך כלל באופן אוטומטי על ידי המחשב. אנחנו עלולים להבחין בזה שהדיסק מחזיק יותר, או שדפי האינטרנט מציגים מהר יותר אבל המחשב בעצם מעבד יותר (עובד יותר).

הומצאו שיטות רבות לדחיסת נתונים. בפעילות זו השתמשנו בשיטה שמשתמשת בעקרון ההצבעה על מופעים קודמים של רצפים בטקסט, שיטה זו מכונה קידוד "למפל-זיו" או בקיצור "LZ" והיא הומצאה על ידי שני פרופסורים ישראליים בשנות ה-70. ניתן להשתמש בקידוד זה לכל שפה ואף להקטין בחצי את הגודל של הנתונים שאנו "דוחסים". במחשבים האישיים הוא מכונה "ZIP", וכמו כן הוא משמש לתמונות "GIF", כמו גם למודמים במהירות גבוהה. במקרה של מודמים, הקידוד מפחית את כמות הנתונים שצריכים להיות מועברים בקו הטלפון כך שזה עובר הרבה יותר מהר.

שיטות אחרות מתבססות על הרעיון כי אותיות שנמצאות בשימוש בתדירות גבוהה יותר צריכות קידוד קצר יותר מהאחרות (לדוגמה – קוד מורס השתמש בזה), נדבר על שיטה כזאת היום – קידוד האפמן.

קוד האפמן

קוד האפמן הוא שיטה לקידוד סימנים, כגון תווי טקסט, ללא אובדן נתונים. הקוד שייך למשפחה שימושית של קודים המכונה **קודי תחיליות***, ובמשפחה זו הוא הקוד המספק דחיסת נתונים מרבית, כלומר מאחסן את הסימנים במספר מזערי של סיביות. השיטה מתבססת על אורך משתנה לסימנים על פי שכיחותם, כך **שסימן נפוץ יוצג באמצעות מספר קטן של סיביות.** לרוב ניתן לחסוך באמצעות שיטה זו בין 20% ל-90% משטח האחסון. קוד האפמן הוא גרסה כללית יותר של עקרון קידוד הנקרא" קידוד אנטרופיה."

* קוד האפמן הוא קוד **תחיליות**: כל מחרוזת ביטים שמייצגת אות אינה תחילית של מחרוזת המייצגת אות אחרת.
דוגמא: קוד עם מילים {9, 59, 55} עומד בקריטריון קוד תחילית, בעוד קוד עם המילים: {5, 9, 59, 55} לא עומד בקריטריון מאחר ו-5 הוא תחילית גם של 59 וגם של 55.

רקע

הדרך הנפוצה ביותר לייצוג אותיות ומספרים הוא באמצעות טבלת ASCII (American Standard Code for Information Interchange). טבלה זו מבוססת על מחרוזות באורך 7 של 'ביטים', לדומא:

Character	*ASCII Code*
A	100 0001
B	100 0010
C	100 0011
D	100 0100
...	...
...	...
1	011 0001
2	011 0010
3	011 0011
...	...

בטבלה מיוצגים גם סימנים וסמלים מוכרים נוספים כמו: ?@!.# ועוד.

דיון במטרה שיגיעו למסקנה שצריך ליעל

שאלה: כמה סימנים ואותיות שונות ניתן לייצג באמצעות 7 ביטים? תשובה: $2 * 2 * 2 * 2 * 2 * 2 * 2 = 2^7 = 128$

שאלה: ומה לגבי טקסט שיש בו מספר אותיות שכיחות? האם יש שיטה יעילה יותר מקידוד ASCII?
נותנים להם לנסות לחשוב ולהעלות מספר פתרונות, מפה ישר מתחילים את פעילות ההדגמה.

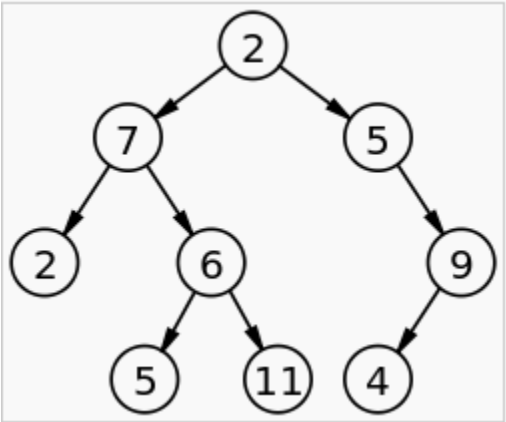
דוגמא פשוטה: (במטרה לעזור לענות על השאלה) נניח שנתונה מחרוזת בת 200 תווים, שמחציתם האות א. בקוד ASCII, שבו לכל תו מוקדשות 8 סיביות, אורכה של מחרוזת זו הוא 1,600 סיביות. נקודד כעת את

התווים בדרך חדשה: האות א תסומן בסיבית יחידה שערכה 0, וכל תו אחר יסומן בקוד ה ASCII-הרגיל שלו, שבתחילתו תתוסף הסיבית 1 (תוספת זו הכרחית, כדי שנוכל להבדיל בין סיבית 0 שמציינת את האות א, ובין סיבית 0 באמצע תו כלשהו). אורך המחרוזת בקוד זה הוא 1,000 סיביות בלבד, שהם 63% מאורך המחרוזת המקורית.

פעילות 3 – מהו עץ בינארי?

עץ בינארי הוא עץ שבו לכל קודקוד יש לכל היותר שני בנים, ולכל קודקוד, פרט לקודקוד מיוחס הנקרא שורש, יש אב יחיד. קודקוד של עץ נקרא גם צומת. ציירו את העץ הבא על הלוח (לא חייבים לצייר את החיצים), וציינו כי קודקוד 2 הינו אב של 5 ו-7, ובהתאמה 5 ו-7 הם בניו של 2. 5 הוא בן ימני של 2, 7 הוא בן שמאלי של 2. עלה בעץ הוא קודקוד בלי בנים.

חשוב להדגיש מי הם הבנים ומה הם סוגי הבנים (ימני ושמאלי) להמשך הפעילות, מאחר והילדים יצטרכו לבנות עצים בעצמם.



נרצה להראות איך בונים עץ בינארי לקוד האפמן: שלב ראשון יהיה: על כל קשת שמאלית שיוצאת מקודקוד (כלומר, בן שמאלי של קודקוד) נכתוב את הספרה 1, ועל כל קשת ימנית שיוצאת מקודקוד נכתוב את הספרה 0.

<u>קידוד חדש:</u>

הוסיפו את הספרות על הקשתות, ועשו מספר דוגמאות – דוגמאות אלה באות לקודד מסלול מסוים (כל המסלולים מתחילים בשורש ומסתיימים בעלה כלשהו). קראו ללוח למתנדבים. עבור המסלולים הבאים:

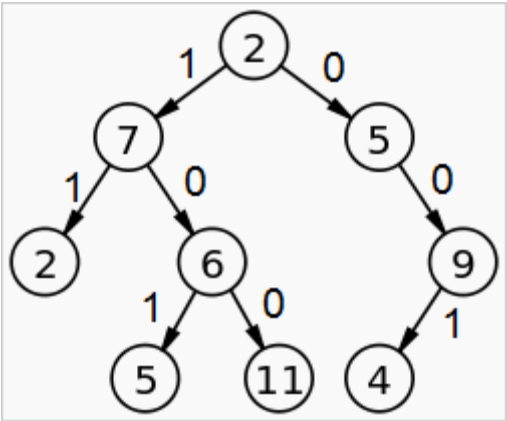
2←5←9←4 הפלט יהיה: 001

2←7←6←5 הפלט יהיה: 101

2←7←2 הפלט יהיה 11.

ניתן לראות כי לא כל המסלולים באותו האורך.

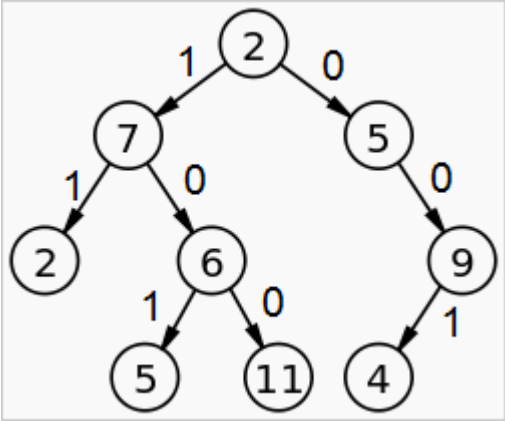
נחלק את דף פעילות 3 כדי שהתלמידים יתרגלו את שיטת קידוד זו.



דף פעילות 3: קידוד מסלולים בעץ בינארי

קודדו את המסלולים הבאים לפי העצים הבינאריים השונים:

1. השלימו את קידוד המסלולים:

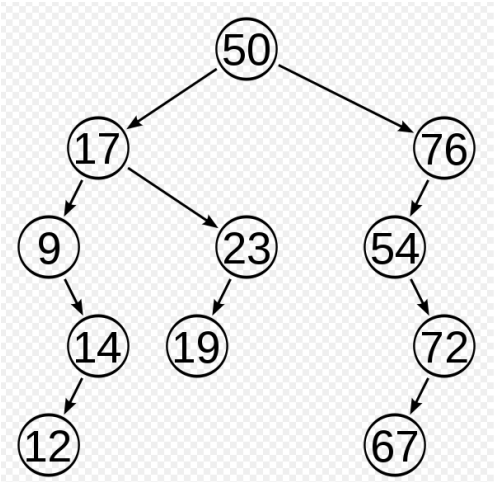


2←5←9←4: _____

2←7←6←11: _____

2←7←1: _____

2. הוסיפו את הספרות '0' ו-'1' על הקשתות כפי שראיתם בכיתה, והשלימו את קידוד המסלולים:

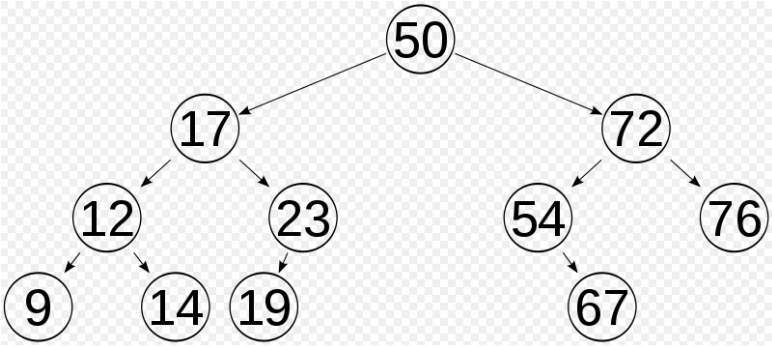


50←17←23←19: _____

50←76←54←72←67: _____

50←17←9←14←12: _____

3. גם כאן, הוסיפו את הספרות על הקשתות והשלימו את קידוד המסלולים:



50←72←76: _____

50←17←23←19: _____

50←17←12←9: _____

פעילות 4 - הדגמת השיטה

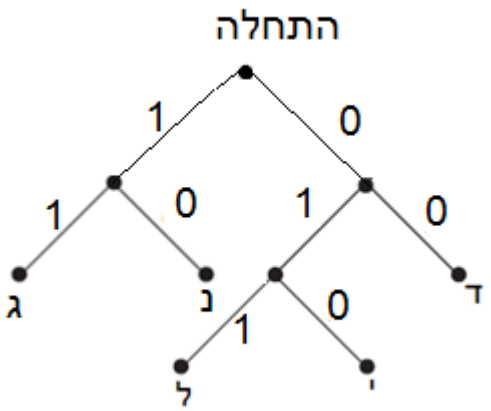
למדנו מהו עץ בינארי, נמשיך בלימוד שיטת קידוד האפמן (בהמשך הדף נלמד איך בונים עץ בינארי עבור הקידוד).
שיטת הקידוד של ASCII, בו כל אות מיוצגת על ידי 7 ביטים, אינה יעילה במיוחד במקרים בהם הזיכרון מוגבל (מחשב), או כשאנו מנסים לדחוס מידע (כמו שרוב התוכנות שמעבירות מידע בין מחשב למחשב עושות – כמו נניח שליחת קובץ במייל).

כאשר שולחים מידע, למשל מייל, הרבה יותר יעיל להשתמש בקוד שייצג **בפחות ביטים אותיות שכיחות** בטקסט, וייצג **ביותר ביטים אותיות נדירות** בטקסט (מספר קטן יותר של ביטים = יותר יעיל). זה הבסיס לקוד האפמן, אשר פותח בשנת 1952 באוניברסיטת MIT.

נדגים את השיטה עם הטקסט הבא: **גנן גידל דגן**

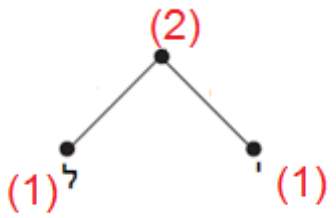
נבנה טבלה ונמלא אותה בעזרת עץ בינארי – אותו נבנה מלמטה למעלה.

אות	מספר הופעות בטקסט	קוד	אורך בקידוד האפמן	אורך בקידוד ASCII
ג	3	11	2	7
נ	3	10	2	7
ד	2	00	2	7
י	1	010	3	7
ל	1	011	3	7

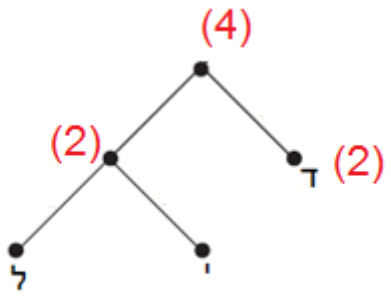


<u>איך בונים את העץ עצמו?</u>

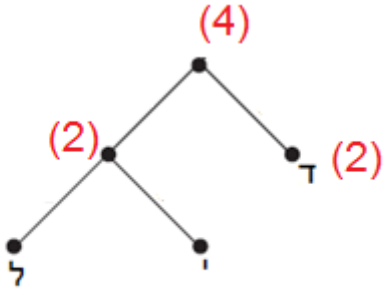
לפי מספר ההופעות של כל אות בטקסט:
בכל שלב בוחרים את שתי האותיות שמספר הופעתן הינו הנמוך ביותר בטקסט, במקרה שלנו י' ו-ל'. שמים אותם תחת אותו עץ וזוכרים בשורש את מספר הופעתן:

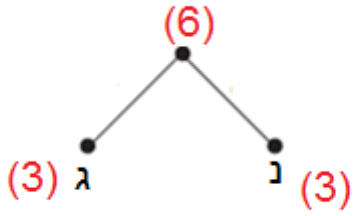


כעת שוב, בוחרים את שתי האותיות עם מספר הופעות נמוך ביותר (לוקחים בחשבון את י' ו-ל' בתור אות אחת!), ושוב מחברים – כלומר שמים תחת אותו אב את שתי האותיות. בשלב השני שלנו נשים את ד' יחד עם ל' ו-י':

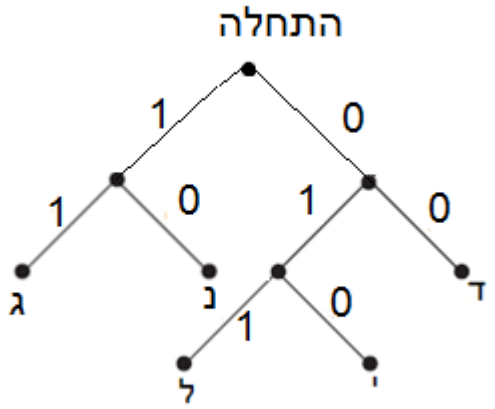


ממשיכים בתהליך, כעת נחבר את נ' ו-ג' והיער שלנו יראה כך:





שלב אחרון, נחבר את שני העצים שלנו ונקבל (עם הוספת הספרות על הקשתות):



אם נחשב את ההבדל בכמות הביטים בשתי השיטות:

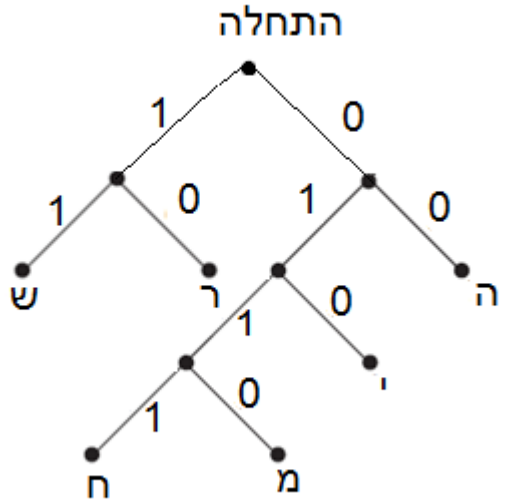
- ASCII: יש לנו 10 אותיות (נתעלם מהרווחים) ומאחר ואנו מקצים 7 ביטים לאות נצטרך **7*10=70** ביטים.
- האפמן: נחשב לפי האורך שהקצנו בטבלה: **3*2+3*2+2*2+1*3+1*3= 22** ביטים! הבדל משמעותי.

כעת ניתן לחלק את דף פעילות מספר 2, שישלימו לבד עבור: **שרה שרה שיר שמח**.
פתרון לדוגמא (לא יחיד, יתכן פתרון נכון אך לא זהה אם מסדרים שונה את האותיות בעלי העץ הבינארי).

אות	מספר הופעות בטקסט	קוד	אורך בקידוד האפמן	אורך בקידוד ASCII
ש	4	11	2	7
ר	3	10	2	7
ה	2	00	2	7
י	1	010	3	7
מ	1	0110	4	7
ח	1	0111	4	7

נזכיר את השיטה:

- ח' ו-מ' יהיו תחת אותו שורש (כעת משקלם 2)
- אליהם נחבר את י' (משקל י' 1)
- אליהם נחבר את ה' (משקל ה' 2 ומשקל העץ שבנינו עד כה הוא 3 – סה"כ 5)
- נחבר את ר' (משקל 3) עם ש' (משקל 4)
- נחבר את שני העצים.

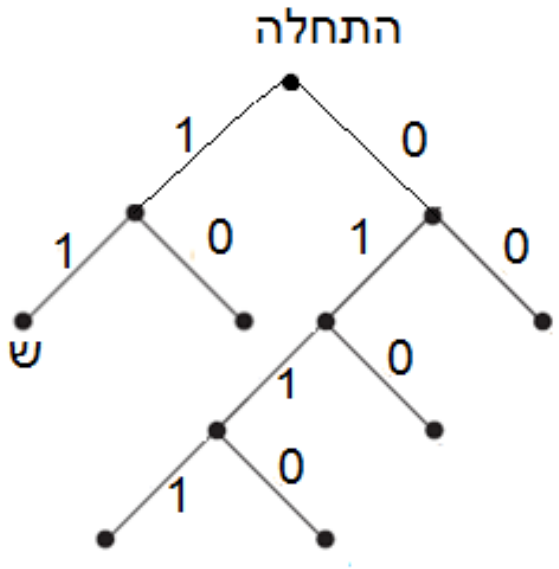


דף פעילות 4: קידוד בשיטת האפמן

עכשיו תורכם! קדדו את המשפט הבא כפי שנעשה בכיתה:

שרה שרה שיר שמח

אות	מספר הופעות בטקסט	קוד	אורך בקידוד האפמן	אורך בקידוד ASCII
ש	4	11	2	7
ר				
ה				
י				
מ				
ח				



כעת חשבו כמה ביטים אנו צריכים לייצג את המשפט: שרה שרה שיר שמח בכל אחת משיטות הקידוד:

- קידוד ASCII: _____

- קידוד האפמן: _____

פעילות 5 – פענוח חזרה

כעת, נרצה לפענח קידוד כלשהו חזרה למילה, בעזרת טבלת הקידוד שלנו (שבנינו עכשיו למשפט "שרה שרה שיר שמח"). על מנת לעשות זאת יחד עם כולם **מומלץ לישר קו** מבחינת טבלת הקידוד, מאחר והן **אינן ייחודיות**, ואם לא יישרו קו אז הם יקבלו מילים אחרות..

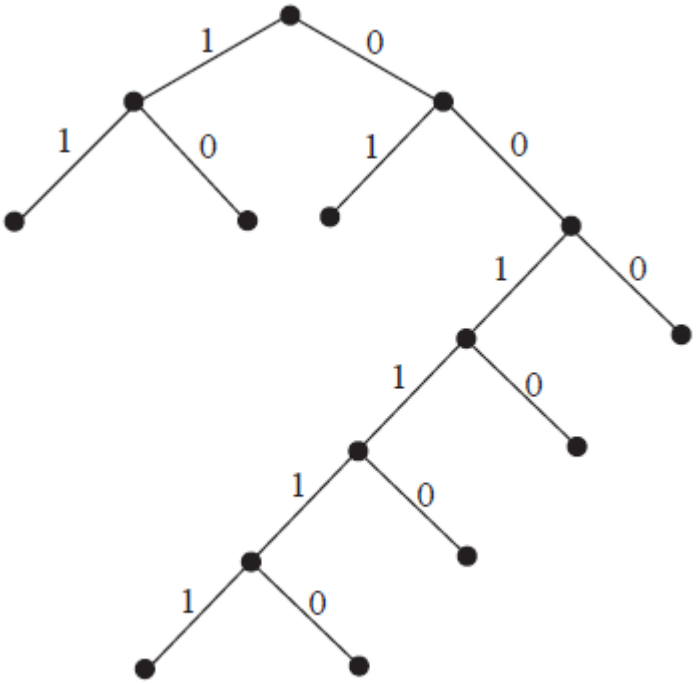
- 10 0111 0110 (מחר)
- 010 010 00 (היי)
- 10 0111 11 (שחר)

תכתבו על הלוח את המספרים ברצף. אמורים לזהות את האותיות בקריאה **משמאל לימין** כי מדובר בקוד **תחיליות** (והוא נקרא משמאל לימין). לאחר פענוח האותיות, נקרא את המילה מימין לשמאל. מה לעשות שעברית הפוכה מכל השפות..? ☺

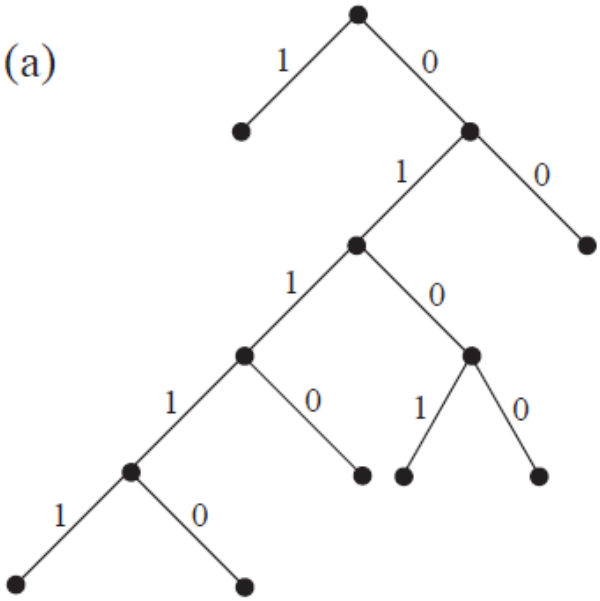
פעילות 6 – הסבר על הבדלים בעצים

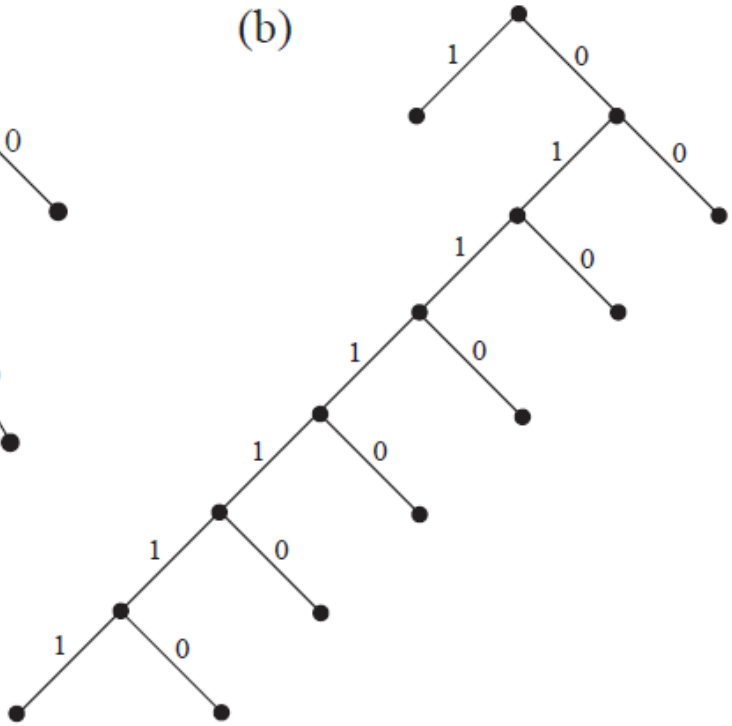
ככל ששכיחות האותיות בטקסט מסוים מאוזנת יותר – כלומר, נניח יש לנו טקסט מסוים עם 4 אותיות וכולן חוזרות על עצמן 3 פעמים בטקסט – כך יהיה העץ שלנו **שלם** יותר, כלומר מאוזן, רוב האותיות ייצוגו על ידי **מספר זהה של ביטים**. לעומת זאת, בטקסט בו יש נניח מופיעה אות מסוימת 100 פעמים, לעומת 3 אותיות נוספות שמופיעות פעם אחת, העץ שלנו יראה פחות מאוזן.

שאלה: כמה עצים שונים ניתן לצייר עבור קוד עם 8 אותיות?
לא חייבים לצייר את כולם, אבל הנה אחד לדוגמא:

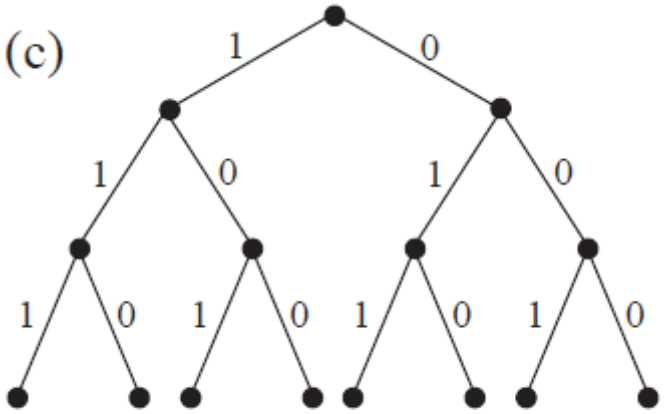


תנו לילדים לחשוב על עצים שונים שייצגו 8 אותיות, ונסו ביחד לחשוב על מה מייצגים סוגי עצים אלה. הנה כמה לדוגמא:





עץ שלם, בו רוב האותיות מופיעות אותו מספר פעמים:



דף פעילות 5: פיתוח קוד האפמן

פתחו קוד האפמן עבור שפה בה המילים היחידות הן (היעזרו בעץ הבינארי שבדף הבא):

כלב, רכב, כיף, פוף, ילד, דחף, ברח, כדור, דרדס

אות	מספר הופעות בטקסט	קוד	אורך בקידוד האפמן	אורך בקידוד ASCII

שלחו קוד לחבר! ☺

העתיקו את קידוד האותיות בלבד:

אות	קוד

וכאן תיצרו מילים מקודדות (כלומר מביטים). לאחר מכן, העבירו לחבר שיקודד את הביטים חזרה למילים:

1. _____

2. _____

3. _____

דף פעילות 5: פיתוח קוד האפמן - המשך

תמלאו את עמודת ה"קוד" בטבלה בעזרת העץ הבינארי. איך יראה העץ הבינארי שלכם? מאוזן יחסית? מדורג? שימו לב שאתם מתייחסים למספר הופעות האותיות במילים.