ערמה בינארית: עץ בינארי מלא המיושם במערך. אני חביב יותר מכל תת העץ שלי. אם אני באינדקס i אז אבא ב- $\frac{i}{2}$, והבנים ב- $2i, 2i+1$.

- פעולה בסיסית ($Heapify$): משווים בין שני האחים, המנצח נגד האבא ואם הבן מנצח מחליפים. $O(1)$

1. <u>מצא חביב:</u> החביב הוא השורש – $O(1)$
2. <u>הוצא חביב:</u> נשים את האחרון במקום הראשון ומשם פעולה בסיסית עד שמתקיימת חביבות, $O(\log n)$
3. <u>הוסף רשומה:</u> נוסיף לסוף המערך ואז פעולה בסיסית, $O(\log n)$
4. <u>בנה ל- n רשומות:</u> מכניסים את כולם לעץ - $O(n)$ ומשם מבצעים מלמטה למעלה פעולות בסיסיות, סה"כ $O(n)$.
5. <u>איחוד:</u> אם $n \gg m$ נוסיף אחד אחד לגדולה, $O(m \cdot \log n)$. אחרת נוסיף את כולם לערמה חדשה, $O(n+m)$.
6. <u>שיפור חביבות:</u> נשפר חביבות ופעולה בסיסית, $O(\log n)$.

ערמה בינומית: כל עץ מוגדר רקורסיבית. B_0 הוא צומת בודד, ולשאר:

$$i \ge 1 \Rightarrow B_i = Union(B_{i-1}, B_{i-1})$$

בעץ B_k יש 2^k צמתים, גובהו k, לשורש k ילדים, וברמה ה- i יש $\binom{k}{i}$ צמתים. ערמה בינומית היא יער של עצים בינומיים מסודרים לפי הגודל.

ייצוג בינארי!

תכונת החביבות מתקיימת לכל עץ בנפרד, רשימה מקושרת של השורשים (יש $O(\log n)$ עצים).

גובה הערמה הוא לכל היותר $\log n$.

<u>פונקציית הפוטנציאל</u> של המבנה היא <u>מספר העצים</u> בערמה.

1. <u>מצא חביב:</u> החביב מבין השורשים, $O(\log n)$.
2. <u>הוצא חביב:</u> מוצא ומוציא חביב, מאחדים את הבנים עם הערמה, $O(\log n)$.
3. <u>הוסף רשומה:</u> כמו איחוד עם B_0, $O(\log n)$.
4. בנה ל- n רשומות: כמו מונה בינארי, $O(n)^*$.
5. <u>איחוד:</u> כמו חיבור מספרים בינאריים, $O(\log n)$.
6. <u>שיפור חביבות:</u> פעולה בסיסית בתוך העץ, $O(\log n)$.
   - <u>אין כיוון?</u> חשבו על מספרים בינאריים.

ניתוח לשיעורין:

1. <u>שיטת הצבירה:</u> נסכום את עלות הפעולות באופן ישיר.
2. <u>שיטת הקופונים:</u> ניתן לכל פעולה מספר כלשהו של קופונים, אם לאחר כל הפעולות הצלחנו לשלם עם הקופונים שלנו הצלחנו.
3. <u>שיטת הפוטנציאל:</u> נבנה פונקציית פוטנציאל מתאימה ונחשב את $\hat{c}_i$: $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

שיטת האב: $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)\ ;\ d = \log_b a$

1. אם $\exists c < d : f(n) = O(n^c)$ אז $T(n) = \Theta(n^d)$
2. אם $f(n) = \Theta(n^d \cdot \log^k n)$ אז $T(n) = \Theta(n^d \cdot \log^{k+1} n)$
3. אם $\exists c > d : f(n) = \Omega(n^c)$ וגם $\exists k < 1 :\ a \cdot f\left(\frac{n}{b}\right) \le k \cdot f(n)$ אז $T(n) = \Theta(f(n))$

פעולה	ערמה בינארית	ערמה בינומית	ערמת פיבונאצ'י
מצא חביב	$O(1)$	$O(\log n)$	$O(1)$
הוצא חביב	$O(\log n)$	$O(\log n)$	$O(\log n)$
הוסף רשומה	$O(\log n)$	$O(\log n)$	$O(1)$
בנה ל- n	$O(n)$	$O(n)^*$	$O(n)$
איחוד	$O(n+m)$	$O(\log n)$	$O(1)$
שיפור חביבות	$O(\log n)$	$O(\log n)$	$O(1)^*$

מיון	סיבוכיות זמן
מקסימום	$O(n^2)$
בועות	$O(n^2)$
הכנסה	$O(n^2)$
ערימה	$O(n \cdot \log n)$
מהיר	$O(n^2), avg\ O(n \cdot \log n)$
מיזוג	$O(n \cdot \log n)$
ספירה	$O(n+k)$
בסיס	$O(d \cdot (n+b))$

- לנוחיות בלבד, $O(f(n))^*$ – עלות משוערכת $f(n)$.

<u>בחירה במיון המתאים עבור טווח ערכים k:</u>

- אם $k = O(n)$ מיון ספירה
- אם $k = n^k$ מיון בסיס עם סיבובים, כל עוד $k = O(\log n)$
- מספר מפתחות קטן מאוד? מיון הכנסה עם נציגים

ערמת פיבונאצ'י: יער של עצים חביבים, מצביע min לשורש החביב.

צמתים אפורים ושחורים. ניתן לנתק בן רק פעם אחת ובניתוק משחירים. כאשר נרצה להוציא בן של צומת שחור נעביר את השחור לרשימת העצים ונשחיר את אביו (רקורסיבית).

$$t(H) = \#of\ trees$$

$$m(H) = \#of\ marked\ (black)\ nodes$$

$$D(n) = \text{מספר ילדים מקסימלי בעץ} = O(\log n)$$

<u>פונקציית הפוטנציאל</u> של המבנה: $\Phi_i = t_i + 2 \cdot m_i$

1. <u>מצא חביב:</u> הולכים לאן שמצביע min, $O(1)$.
2. <u>הוצא חביב:</u> מוציאים חביב, מתקנים את הערמה ומעדכנים min עולה באמת $O(D(n)+t(H))$, $O(D(n)) = O(\log n)$.
3. <u>הוסף רשומה:</u> עץ חדש משמאל ל- min, $O(1)$.
4. <u>בנה ל- n רשומות:</u> הוסף n פעמים, ולכן $O(n)$.
5. <u>איחוד:</u> שרשור הרשימות המקושרות, $O(1)$.
6. <u>שיפור חביבות:</u> כל עוד אבא מסומן נתק, השחר אבא והוסף כעץ חדש. בסוף עדכן min. מספר האבות השחורים ברצף c, ואז $O(c)$. $O(1)^*$.

<u>פונקציות פוטנציאל לדוגמא:</u>

מונה בינארי: מספר ה- 1ים.
מערך דינמי: $\Phi_i = 2 \cdot n_i - s_i$
(n_i – כמה איברים במערך, s_i – גודל המערך)
ערמה בינומית: מספר העצים.
ערמת פיבונאצ'י - $\Phi_i = t_i + 2 \cdot m_i$
עץ Splay: $\Phi = \sum r(v)$
עץ קרטזי: אורך השדרה הימנית

עץ חיפוש בינארי: עץ בינארי לא מאוזן, כל צומת חביבה יותר מתת העץ השמאלי שלה וחביבה פחות מהימני. פעולות:

1. מצא רשומה: אם גדול מדי לך ימינה, קטן מדי שמאלה עד שמצאנו / עלה, $O(h)$.
2. הוסף רשומה: נסה למצוא, כך הגענו לאבא המיועד שהוא עלה – הוסף לאבא, $O(h)$.
3. הוצא רשומה: נסה למצוא. לא מצאנו – סיימנו, עלה – מחק, בן יחיד – חבר סבא לנכד, שני בנים – מחק החלף בעוקב / קודם, $O(h)$.

סיבוכיות כל הפעולות $O(h) = O(n)$.

עץ אדום-שחור: עץ בינארי מאוזן. שחור ו-$null$ שחורים, לאדום בנים שחורים.

- בכל מסלול מצומת לעלה אותו מספר של צמתים שחורים.

1. מצא רשומה: כמו בעץ חיפוש בינארי, $O(\log n)$.
2. הוסף רשומה: מצא, הוסף למקום, צבע באדום. אם הורה שחור סיימנו, אחרת סוג של פיצול ומסדרים.
3. הוצא רשומה: מצא, אם עלה אדום קל. עלה שחור – כל הסיפור. אם צומת פנימית: החלף בעוקב / קודם – החלפנו בעלה (חזור למקרה הוצא עלה) או בצומת עם בן יחיד (שחייבת להיות שחורה ועם בן אדום – הוצא את העוקב / קודם והשחר את הבן)

עץ קרטזי: השורש הוא אינדקס המינימום. הבן השמאלי הוא המינימום של תת המערך השמאלי, וסימטרי ימינה. ה-LCA של שני צמתים בעץ הוא התשובה ל-RMQ של האינדקסים.

- פונקציית הפוטנציאל היא אורך השדרה הימנית, שקול למספר האחדים במונה בינארי.

עץ דרגות: הרחבה של עצי חיפוש. נוסיף שדה של גודל תת העץ ונענה על שתי שאילתות נוספות ב-$O(\log n)$.

1. Select(i): יוחזר האיבר ה-i הכי קטן.
2. Rank(x): יוחזר הדירוג של x (המיקום במערך ממוין).

בעזרת הרחבה זו ניתן לתחזק שאילתות נוספות.

עץ 2-3-4: עץ כללי מאוזן, בצומת 1-3 רשומות ו2-4 (או 0) בנים. הרשומות בצומת חביבות יותר מכל רשומה ותת עץ שמשמאלה, ופחות מימינה. כל העלים באותה רמה.

הפעולה הבסיסית – פיצול, הכנס N לרשומה מלאה A, B, C: (עלות $O(h)$)

- העלה את B לאבי הצומת ופצל A, C לשני צמתים נפרדים.
  a. $N < B \Leftarrow$ הכנס את N ל-A
  b. $N > B \Leftarrow$ הכנס את N ל-B
- אם אין מקום ב- B הפעל פיצול על ההורה.

1. מצא רשומה: אם נמצא ברשומה סיים, חפש בתת העץ המתאים לפי היחס בין הרשומות למפתח, אם עלה סיים, $O(h)$.
2. הוסף רשומה: נסה למצוא, כך הגענו לאבא המיועד שהוא עלה, אם יש מקום הוסף, אחרת הפעל פיצול, $O(h)$.
3. הוצא רשומה: נסה למצוא. צומת פנימי – החלף בקודם / עוקב. עלה עם יותר מרשומה אחת -- הוצא. עלה עם רשומה אחת – לב הבעיה (הוצא ואזן את העלים), $O(h)$.

לב הבעיה:

1. עזרה מאח: אם לאח יותר מרשומה אחת. הרשומה המתאימה מהאח עולה לשורש, השורש מחליף את זו שיוצאה.
2. עזרה מהורה: אם אין עזרה מאח ולהורה יותר מרשומה אחת. הרשומה המתאימה מההורה יורדת לאח ומחליפה את זו שיוצאה.
3. הגדלת הבעיה: אם אין עזרה מאח / הורה. נעלה את האח להורה וכך רמת עלי האח תהיה שווה לרמת העלים של זו שיוצאה.

מיון או דברים עם הרבה 2-3-4? נסו לשפוך למערך/מערכים ב-in-order.

עץ מילים: מעולה כאשר המפתחות מורכבים מתווים. כל קשת מייצגת מעבר של אות. אולי לנסות עץ סיפות?

עץ Splay: עץ בינארי לא מאוזן. בכל חיפוש נעלה את המחופש (או איפה שנתקענו - אם לא נמצא) לשורש העץ על ידי רוטציות כפולות שיהפכו את העץ ללא פחות מאוזן משהיה. $s(x)$ - גודל תת העץ של הצומת x, $r(x) = \log s(x)$. פונקציית הפוטנציאל:

$$\Phi = \sum_{v \in T} r(v) = \sum_{v \in T} \log s(v) = \log\left(\prod_{v \in V} s(v)\right)$$

$$\hat{c}_i = 3(\log n - \log|T_x|)$$

- כאשר $|T_x|$ הוא גודל תת העץ של הצומת שחיפשנו.