1

1. נשתמש בשיטת האב כאשר $a = 1, b = 4, f(n) = \sqrt{n}$. עבור $\epsilon = \frac{1}{2}$ מתקיים $\sqrt{n} = \Omega(n^{log_4 1+\epsilon}) = \Omega(\sqrt{n})$ וכן אם $c = \frac{1}{2}$ אזי $\sqrt{\frac{n}{4}} = \frac{\sqrt{n}}{2} \leq \frac{\sqrt{n}}{2}$.
זה מתאים למקרה שלישי ולכן $T(n) = \Theta(\sqrt{n})$.

2. מקרה ראשון של שיטת האב: $\lg^2 n = O(n^{\log_5 5-\frac{1}{2}}) = O(\sqrt{n})$ (למשל כי $\lim_{n\to\infty} \frac{\lg^2 n}{\sqrt{n}} \to 0 \implies \lg^2 n = o(\sqrt{n})$) לכן $T(n) = \Theta(n)$.

3. מקרה שני של שיטת האב: $f(n) = n + \frac{n}{\lg n} = \Theta(n) = \Theta(n^{\log_6 6})$ לכן $T(n) = \Theta(n \lg n)$.

4. מקרה שלישי של שיטת האב: $f(n) = n^{\frac{3}{2}} = \Omega(n^{\log_4 4+\frac{1}{4}}) = \Omega(n^{\frac{5}{4}})$ ואם נבחר $c = \frac{1}{2}$ נקבל $4(\frac{n}{4})^{3/2} = \frac{1}{2}n^{3/2}$ ולכן $T(n) = \Theta(n^{3/2})$.

5. נסתכל איך נראית נוסחת הנסיגה לאחר i איטרציות:

$$\begin{aligned} T(n) &= \frac{3}{2} \cdot T(n^{1/2}) + \lg^2 n = \frac{3}{2}(\frac{3}{2} \cdot T(n^{1/4}) + \frac{\lg^2 n}{2}) + \lg^2 n \\ &= \frac{3^2}{2^2} \cdot T(n^{1/4}) + \frac{3}{4}\lg^2 n + \lg^2 n = \cdots \\ &= \left(\frac{3}{2}\right)^i \cdot T(n^{1/2^i}) + \lg^2 n \sum_{k=0}^{i-1} (\frac{1}{3})^k \end{aligned}$$

עבור $i = \lg\lg n$ מתקיים $n^{1/2^i} = 2$ ונקבל:

$$\begin{aligned} T(n) &= \overbrace{\left(\frac{3}{2}\right)^{\lg\lg n}}^{=\frac{\lg^{\lg 3} n}{\lg n}} \cdot T(n^{1/2^{\lg\lg n}}) + \lg^2 n \underbrace{\sum_{k=0}^{\lg\lg n-1} (\frac{1}{3})^k}_{<2} \\ &= O(\lg n) \cdot T(2) + \lg^2 n \cdot O(1) \end{aligned}$$

ובסה"כ $T(n) = \Theta(\lg^2 n)$.

6. בדומה ל-5, לאחר i איטרציות נקבל:

$$T(n) = n^{1/2}T(n^{1/2}) + n\lg n = \cdots = n^{\frac{2^i-1}{2^i}} \cdot T(n^{1/2^i}) + \frac{2^i - 1}{2^{i-1}}$$

עבור $i = \lg\lg n$ מקבלים:

$$T(n) = n^{\frac{\lg n-1}{\lg n}} T(2) + 2 \cdot \frac{\lg n - 1}{\lg n} \cdot n\lg n = \Theta(n) + \Theta(n\lg n) = \Theta(n\lg n)$$

2

א

1. במקרה זה נוסחת הנסיגה היא $T(n) = T(n/2) + \Theta(1)$ וידוע שפתרונה הוא $\Theta(\lg n)$.[1]
2. בכל קריאה רקורסיבית עלינו להעתיק מערך בגודל N לכן נוסחת הנסיגה היא:
$$T(n) = T(n/2) + \Theta(N) = \cdots = T(n/2^i) + i \cdot \Theta(N)$$
כאשר $i = \lg n$ נקבל $T(n) = T(1) + \lg n \cdot \Theta(N) = \Theta(n \lg n)$.
3. כאן עלות ההעתקה היא כגודל תת המערך שבו אנו מחפשים, כלומר $T(n) = T(n/2) + \Theta(n)$ ולאחר i איטרציות נקבל:
$$\begin{aligned} T(n) &= T(n/2) + \Theta(n) \\ &= T(n/4) + \Theta(n/2) + \Theta(n/4) = \cdots \\ &= T(n/2^i) + \sum_{k=0}^{i-1} \Theta(\frac{n}{2^k}) \\ &= T(n/2^i) + \Theta\left(\sum_{k=0}^{i-1} \frac{n}{2^k}\right) \end{aligned}$$
כאשר $i = \lg n$ נקבל:
$$T(n) = T(1) + \Theta\left(\sum_{k=0}^{\lfloor \lg n \rfloor - 1} \frac{n}{2^k}\right) = T(1) + \Theta(n \underbrace{\sum_{k=0}^{\lfloor \lg n \rfloor - 1} \frac{1}{2^k}}_{\leq 2}) = \Theta(n)$$

ב

1. אין שינוי מנוסחת הנסיגה הידועה של מיון מיזוג במקרה הזה, $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \lg n)$.[2]
2. $T(n) = 2T(n/2) + \Theta(n) + \Theta(N)$, לאחר i איטרציות נקבל:
$$T(n) = 2^i T(n/2^i) + \sum_{k=0}^{i-1} 2^k \Theta(n/2^k) + \sum_{k=0}^{i-1} 2^k \Theta(N)$$
כאשר $i = \lg n$ נקבל:
$$T(n) = \Theta(n) + \Theta(n \lg n) + \Theta(N) \underbrace{\sum_{k=0}^{\lfloor \lg n \rfloor - 1} 2^k}_{= \frac{2^{\lfloor \lg n \rfloor} - 1}{2-1} = \Theta(n)} = \Theta(n \cdot N)$$
כלומר $T(n) = \Theta(n^2)$.
3. עלינו להוסיף $\Theta(n)$ לנוסחה מסעיף 1 אבל $\Theta(n) + \Theta(n) = \Theta(n)$ ולכן נקבל את אותה נוסחת נסיגה שפתרונה $\Theta(n \lg n)$.

[1] זה גם מוכח במדריך הלמידה בעמ' 17.
[2] כפי שמוכח בספר בעמ' 29-31.

3

להלן הגרסה החדשה של השגרה, תוך ההנחה ש-$A[0] = \infty$.

```
Heap-Increase-Key(A, i, key)
1: if key < A[i] then
2:     error "new key is smaller than current key"
3: A[i] ← key
4: while A[Parent(i)] < A[i] do
5:     Swap(A[i], A[Parent(i)])
6:     i ← Parent(i)
```

נוכיח את נכונות השגרה בעזרת שמורת הלולאה:

שמורת לולאה: בתחילת כל איטרציה של לולאת ה-**while** שבשורות 4-6, המערך $A[1..\text{heap-size}[A]]$ מקיים את תכונת ערמת המקסימום, למעט יוצא מן הכלל אפשרי אחד: $A[i]$ עשוי להיות גדול מ-$A[\text{Parent}(i)]$.

אתחול: לפני הכניסה לאיטרציה הראשונה, השמנו ב-$A[i]$ את key כאשר שאר איברי המערך נשארו אותו הדבר. כלומר פרט אולי ל-$A[i], A[\text{Parent}(i)]$ (אם $key > A[\text{Parent}(i)]$) כל איברי המערך מקיימים את תנאי הערמה.

תחזוקה: בכל איטרציה אנחנו מחליפים את $A[i]$ ו-$A[\text{Parent}(i)]$ כך שמתקיים $A[\text{Parent}(i)] \geq A[i]$ ולכן שני אלה מקיימים את תכונת הערמה. לאחר שורה 6, המערך מקיים את תכונת הערמה, פרט אולי ל-$A[i], A[\text{Parent}(i)]$.

סיום: בסיום הלולאה הגענו לאינדקס i המקיים $A[i] \leq A[\text{Parent}(i)]$ ולכן שני אלה מקיימים עכשיו את תכונת הערמה ובכלל שאר איברי המערך גם כן, פרט ל-$A[0]$ שמשמש כזקיף ואינו חלק מהערמה.

4

א

אם מסתכלים על ערימה בינארית כעל עץ אז העץ הוא כמעט שלם, כלומר פרט אולי לרמה האחרונה שמלאה משמאל עד לנקודה כלשהי שאר הרמות מלאות לחלוטין. כלומר לכל צומת z, אם יש לו בן ימני אזי יש לו גם בן שמאלי ולכן לא ייתכן שהמסלול הקצר ביותר לצומת עם פחות משני בנים יימצא בתת-העץ השמאלי של z. לפיכך לכל z מתקיים $\text{npl}[\text{left}[z]] \ge \text{npl}[\text{right}[z]]$ ולכן הערמה היא שמאלית.

ב

נוכיח באינדוקציה על r.

אם $r = 1$ העץ מכיל לפחות צומת אחד ומתקיים $2^r - 1 = 1$.

נניח שהטענה נכונה לכל $k < r$ ונוכיח שהיא נכונה ל-r. תת העץ הימני מכיל $r - 1$ צמתים במסלול הימני לפני ההנחה (פשוט הוצאנו את השורש), לכן מהנחת האינדוקציה הוא מכיל לפחות $2^{r-1} - 1$ צמתים. המסלול הימני של תת העץ השמאלי מכיל לפחות $r - 1$ צמתים (שכן אם נניח שלא אז יהיה קיים צומת z שאינו מקיים $\text{npl}[\text{left}[z]] \ge \text{npl}[\text{right}[z]]$ בסתירה לכך שהעץ הוא עץ שמאלי), ולכן שוב מהנחת האינדוקציה תת העץ השמאלי מכיל לפחות $2^{r-1} - 1$ צמתים.
נסכום את שני תתי העצים עם שורש העץ ונקבל שבעץ לפחות $2\left(2^{r-1}-1\right)+1 = 2^r-1$ צמתים, כנדרש.

מסקנה 1: המסלול הימני בערמה שמאלית בת n צמתים מכיל לכל היותר $\lfloor \lg(n+1) \rfloor$ צמתים.

הוכחה: נניח בשלילה שלערמה בת n צמתים, מסלול ימני בעל $\lfloor \lg(n+1) \rfloor + k$ צמתים כאשר $k \ge 1$. לפי הטענה שהוכחה לעיל, מספר הצמתים בערמה הוא לפחות $2^{\lfloor \lg(n+1) \rfloor + k} - 1 > 2^{\lg(n+1)-1+k} - 1 \ge 2^{\lg(n+1)} - 1 = n$. כלומר מספר הצמתים גדול ממש מ-n בסתירה להנחה.

ג

```
Merge-Leftist(H1, H2)
1:  if H1 = nil then
2:      return H2
3:  if H2 = nil then
4:      return H1
5:  if key[H1] < key[H2] then
6:      Swap(H1, H2)
7:  right[H2] ← Merge-Leftist(H1, right[H2])
8:  parent[right[H2]] ← right[H2]
9:  if left[H2] = nil or npl[right[H2]] > npl[left[H2]] then
10:     Swap(left[H2], right[H2])
11: ▷ H2 npl might have changed, update it
12: npl[H2] ← 1+npl[right[H2]]
13: return H2
```

השגרה עובדת לפי העיקרון המתואר בממ"ן. לפני הקריאה הרקורסיבית, H_1 היא הערמה עם השורש הגדול מבין השניים (כאשר שתי הערמות לא ריקות) ואז ממזגים אותה רקורסיבית עם הבן הימני של H_2 ואת התוצאה שמים כבן ימני שלו.
מלבד שורה 7 כל שאר הפעולות בשגרה לוקחות זמן קבוע. בקריאה הרקורסיבית

בשורה 7 אנחנו יורדים באחד העצים לאורך המסלול הימני ועוצרים בסופו. במקרה הגרוע (שבו שורש העץ הראשון גדול מהשני וקריאה הבאה המצב מתחלף) נרד לסוף המסלול בכל עץ, שהם כאמור לפי מסקנה 1 באורך $\lfloor \lg(n_1+1) \rfloor$ ו-$\lfloor \lg(n_2+1) \rfloor$.
סה"כ נקבל ש-$T(n) = \Theta(\lfloor \lg(n_1+1) \rfloor + \lfloor \lg(n_2+1) \rfloor) = \Theta(\lg n_1 + \lg n_2)$.

ד

Merge-Leftist-Iterative(H_1, H_2)

1: **if** H_1 = **nil then**
2:     **return** H_2
3: **if** H_2 = **nil then**
4:     **return** H_1
5: **if** key[H_2] < key[H_1] **then**
6:     Swap(H_1, H_2)
7: ▷ first pass top to bottom: merge H_1 and H_2, *root* is the last node that was touched
8: **while** $H_1 \neq$ **nil do**
9:     $r \leftarrow$ right[H_1]
10:     **if** r = **nil or** key[H_2] < key[r] **then**
11:         right[H_1] $\leftarrow H_2$
12:         parent[H_2] $\leftarrow$ right[H_1]
13:         $H_2 \leftarrow r$
14:     *root* $\leftarrow H_1$
15:     $H_1 \leftarrow$ right[H_1]
16: $H_1 \leftarrow$ parent[*root*]
17: ▷ second pass bottom to top: fix leftist property and update npl values
18: **while** $H_1 \neq$ **nil do**
19:     **if** left[H_1] = **nil or** npl[right[H_1]] > npl[left[H_1]] **then**
20:         Swap(left[H_1], right[H_1])
21:     npl[H_1] $\leftarrow$ 1+npl[right[H_1]]
22:     *root* $\leftarrow H_1$
23:     $H_1 \leftarrow$ parent[H_1]
24: **return** *root*

השגרה מבצעת שני מעברים על המסלול הימני: במעבר הראשון אנחנו ממזגים את הערמות. התהליך מתבצע על המסלול הימני של הערמה הקטנה שמוחזקת ב-H_1. בכל איטרציה אנחנו יורדים רמה אחת ימינה ובודקים את הצומת מול H_2. בנוסף אנחנו זוכרים את הצומת האחרון שהשתנה כדי שבמעבר השני נרוץ ממנו עד לשורש הערמה הממוזגת ונתקן את תכונת השמאליות במידת הצורך.

לולאת ה-**while** בשורות 8-15 רצה לאורך המסלול הימני של כל עץ במקרה גרוע. הלולאה השניה רצה מסוף המסלול הימני (לאחר שהעצים מוזגו) עד לשורש העץ. וכן כל אחת מהלולאות מבצעת מספר קבוע של פעולות בכל איטרציה. אפשר לתאר את זמן הריצה ע"י הפונקציה:

$$\begin{aligned} T(n) &= \lfloor \lg n_1 \rfloor \cdot \Theta(1) + \lfloor \lg n_2 \rfloor \cdot \Theta(1) + \lfloor \lg(n_1+n_2) \rfloor \cdot \Theta(1) \\ &= \Theta(\lg n_1) + \Theta(\lg n_2) + \Theta(\lg(n_1+n_2)) \\ &= \Theta(\lg n_1 + \lg n_2) \end{aligned}$$

תוצאה זהה לסעיף ג.

ה

ממזגים את הערמה הנתונה עם ערמה חדשה, ששורשה x, ללא ילדים. לפי ג, זמן הריצה הוא $\Theta(\lg n + \lg 1) = \Theta(\lg n)$.

ו

המינימום נמצא בשורש הערמה, לכן כל מה שצריך לעשות זה למזג את תת-העץ השמאלי והימני ולהחזיר את התוצאה. זמן הריצה במקרה זה הוא $\Theta(2\lg n) = \Theta(\lg n)$.